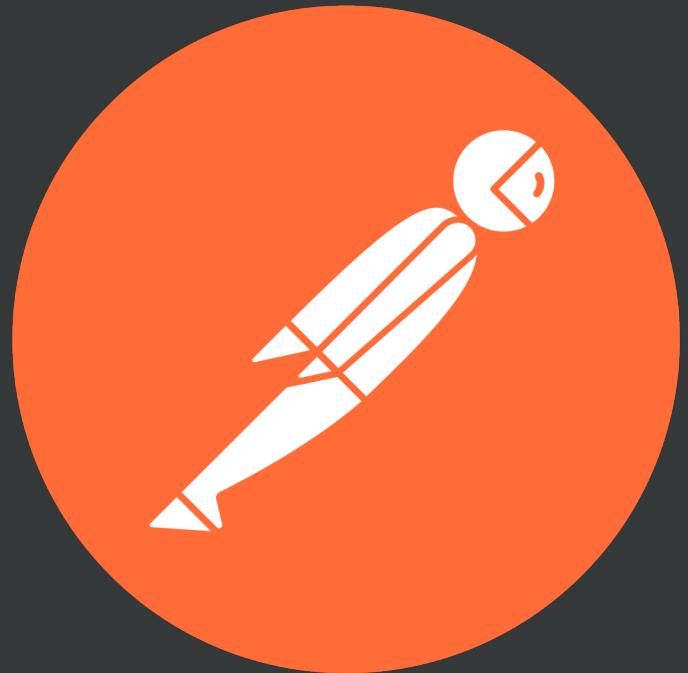




POSTMAN



THE LEGEND OF
P O S T M A N
BREATH OF THE API



Not just the
POSTMAN





POSTMAN

And



Insomnia

API Testing Made Easy

Caleb^{the} Meyer

Senior Software Engineer
Dad of the Millennium (Falcon)
Arbiter of Documentation
Script Ninja
Veteran Mercenary

History
Basics
Requests
Collections
Variables
Testing
Insomnia

Covered Today

Postman started as
a ~~Chrome Extension~~
Side Project

Postman

Home Workspaces Reports Explore Search Postman Sign In Create Account

GET Untitled Request + ... Prod Zone 2

Untitled Request

Save </> Send

GET Enter request URL

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

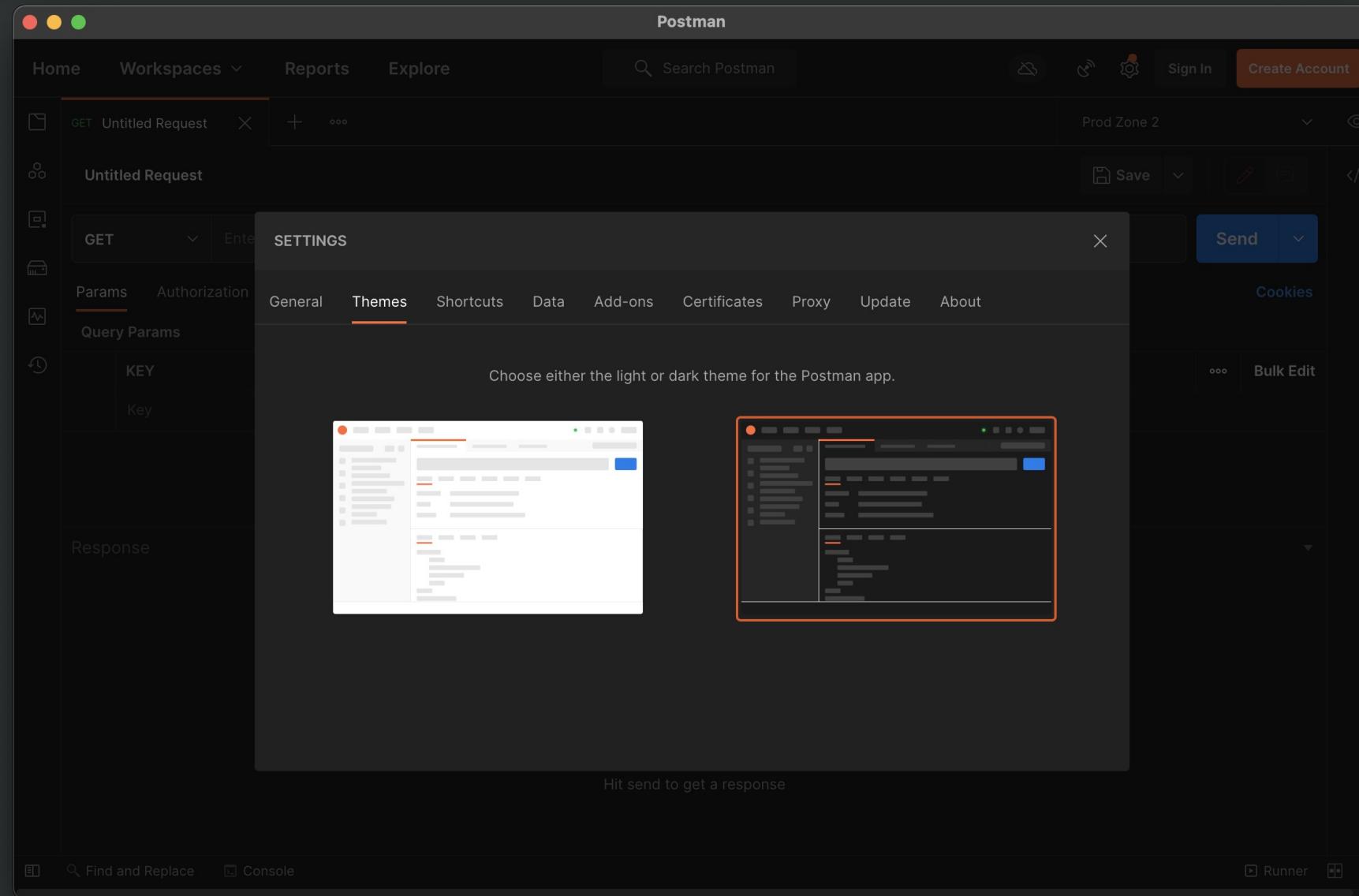
	KEY	VALUE	DESCRIPTION	...	Bulk Edit
	Key	Value	Description		

Response

Hit send to get a response

Find and Replace Console Runner

Too
bright!

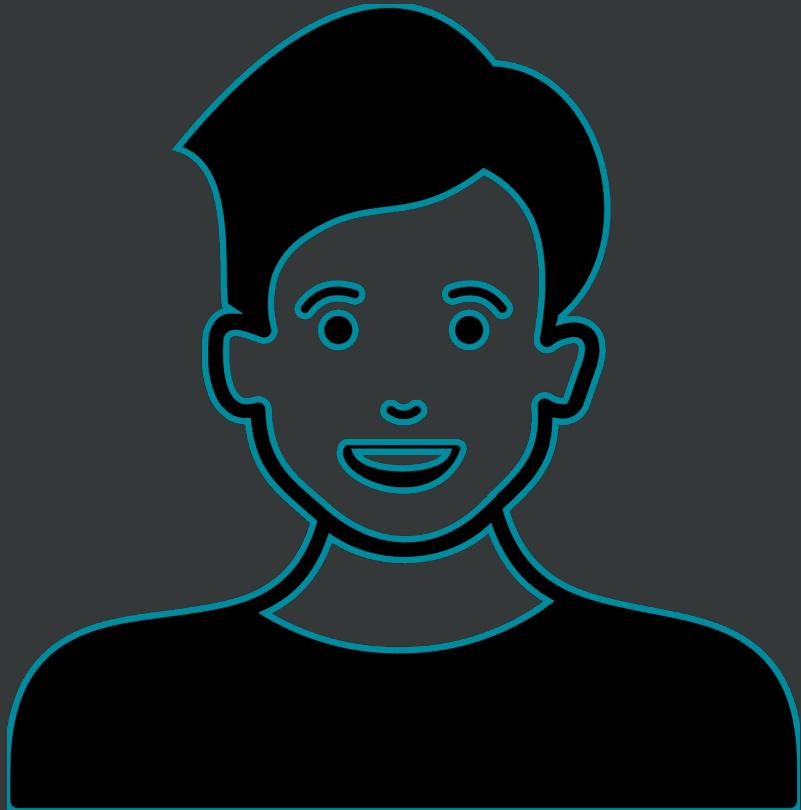


Much
Better!

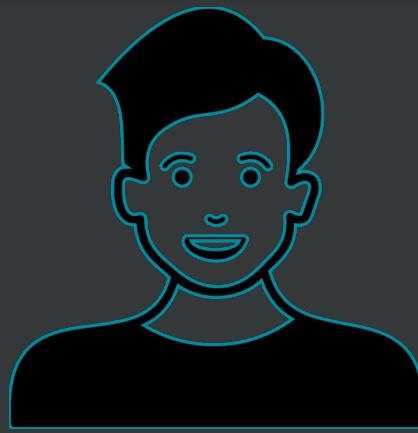
http://
y
p
e
x
t
r
a
n
s
f
e
r

r
a
n
t
o
c
o
l

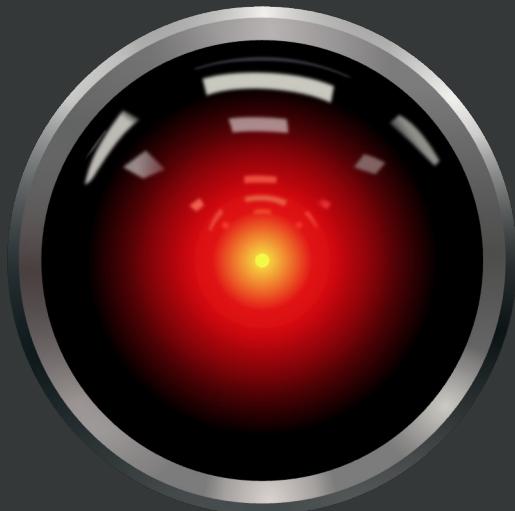




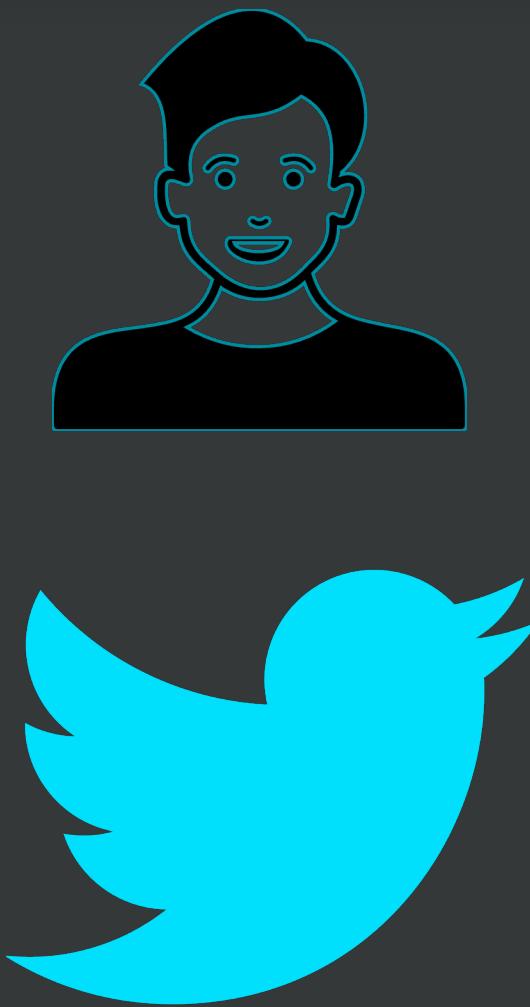
Hello, I want to
GET /tweets



GET /tweets



401 Unauthorized



GET /tweets

200
OK

A screenshot of a tweet from Elon Musk (@elonmusk) dated 7 Aug 2018. The tweet reads: "Am considering taking Tesla private at \$420. Funding secured." It has 15,897 retweets and 87,589 likes.

Elon Musk @elonmusk

Follow

Am considering taking Tesla private at \$420. Funding secured.

12:48 PM - 7 Aug 2018

15,897 Retweets 87,589 Likes

6.3K 16K 88K

GET



GET

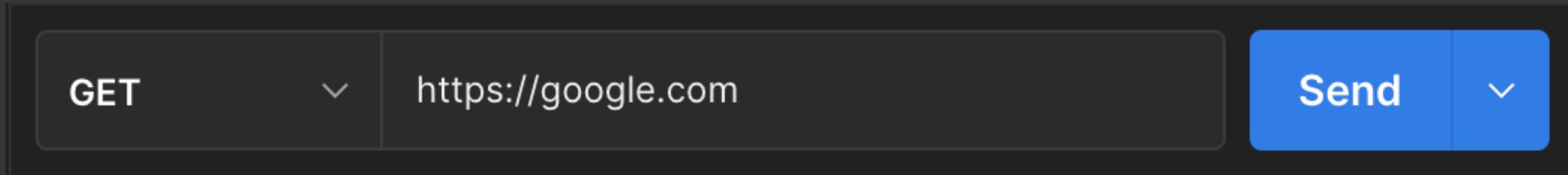
POST

PUT

PATCH

DELETE

HTTP Methods



The URL

Protocol Domain Path Query Parameters

The screenshot shows the Postman interface with the following details:

- Protocol:** GET
- Domain:** https://httpbin.org
- Path:** /get
- Query Parameters:** query=search for data&format=utf-8

The "Params" tab is selected. The "Query Params" table contains the following data:

	KEY	VALUE
<input checked="" type="checkbox"/>	query	search for data
<input checked="" type="checkbox"/>	format	utf-8
	Key	Value

The screenshot shows the Postman interface with the 'Authorization' tab selected. The 'Basic Auth' type is chosen. A note explains that the authorization header will be automatically generated when the request is sent. There are fields for 'Username' and 'Password' with a 'Show Password' checkbox.

Params ● Authorization ● Headers (8) Body Pre-request Script Tests Settings

Type Basic Auth ▾

The authorization header will be automatically generated when you send the request. [Learn more about authorization ↗](#)

Username

Password

Show Password

Authorization



Params ● Authorization Headers (7) Body Pre-request Script Tests Settings

Headers ⚡ Hide auto-generated headers

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> Cache-Control ⓘ	no-cache	
<input checked="" type="checkbox"/> Postman-Token ⓘ	<calculated when request is sent>	
<input checked="" type="checkbox"/> Host ⓘ	<calculated when request is sent>	
<input checked="" type="checkbox"/> User-Agent ⓘ	PostmanRuntime/7.28.0	
<input checked="" type="checkbox"/> Accept ⓘ	*/*	
<input checked="" type="checkbox"/> Accept-Encoding ⓘ	gzip, deflate, br	
<input checked="" type="checkbox"/> Connection ⓘ	keep-alive	

Key Value Description

Params ● Authorization Headers (9) Body ● Pre-request Script Tests Settings

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL JSON ▾

```
1 {  
2   "id": 1,  
3   "name": "Caleb",  
4   "addresses": [  
5     "ICG2",  
6     "ICL7",  
7     "INN.1024.1.8",  
8     "WFH"  
9   ]  
10 }  
11 |
```



Send



Making Requests

The screenshot shows the Postman application interface. At the top, there is a header with 'GET' selected, a URL field containing 'https://httpbin.org/get?query=search for data&format=utf-8', and a 'Send' button. Below the header, there are tabs for 'Params', 'Authorization', 'Headers (9)', 'Body', 'Pre-request Script', 'Tests', and 'Settings'. The 'Params' tab is active, showing two query parameters: 'query' with value 'search for data' and 'format' with value 'utf-8'. The 'Body' tab is also active, displaying the raw JSON response from the request. The response body is as follows:

```
1  [
2   "args": {
3     "format": "utf-8",
4     "query": "search for data"
5   },
6   "headers": {
7     "Accept": "*/*",
8     "Accept-Encoding": "gzip, deflate, br",
9     "Cache-Control": "no-cache",
10    "Host": "httpbin.org",
11    "Postman-Token": "c2ce559d-3163-4ebf-a354-dcc3c2250bf5",
12    "User-Agent": "PostmanRuntime/7.28.0",
13    "X-Amzn-Trace-Id": "Root=1-60db57d0-7453028a5e5747d40e09c9d9"
14  },
15  "origin": "159.140.252.97",
16  "url": "https://httpbin.org/get?query=search for data&format=utf-8"
17 ]
```

The status bar at the bottom indicates a 200 OK status, 68 ms time, and 730 B size. There are also 'Save Response' and filter/search icons.

Responses

Response Status Codes

100 Continue 400 Bad Request 302 Found
204 No Content 404 Not Found 201 Created
401 Unauthorized 503 Service Unavailable
500 Internal Server Error 200 OK 504 Gateway Timeout

Tabs

The screenshot shows the Postman application interface. At the top, there are two tabs: "GET GET with Params" and "POST POST with body". The "POST POST with body" tab is active, indicated by an orange underline. To the right of the tabs are buttons for "Save", "Edit", and "Copy". The main workspace is titled "Postman Presentation / POST with body". Below the title, the method is set to "POST" and the URL is "https://httpbin.org/post". On the right side of this row are "Send" and "Send" buttons. The "Body" tab is selected, indicated by an orange underline. Below this, there are several input options: "none", "form-data", "x-www-form-urlencoded", "raw" (which is selected and highlighted in orange), "binary", "GraphQL", and "JSON" (with a dropdown arrow). The JSON content is displayed in a code editor:

```
1 {  
2   "id": 1,  
3   "name": "Caleb"  
4 }
```

Tabs

The screenshot shows the Postman application interface. At the top, there is a navigation bar with several tabs: 'GET G.', 'POST F X' (which is selected), 'GET U.', '→', '+', '...', and 'Prod Zone 2'. Below the navigation bar, the title 'Postman Presentation / POST with body' is displayed. On the right side of the header, there are buttons for 'Save', '...', 'Edit', and 'Copy'. The main workspace shows a 'POST' request configuration with the URL 'https://httpbin.org/post'. The 'Body' tab is selected, showing the following JSON payload:

```
1  {
2    "id": 1,
3    "name": "Caleb"
4 }
```

Below the body, there are tabs for 'Params', 'Authorization', 'Headers (9)', 'Pre-request Script', 'Tests', 'Settings', 'Cookies', and 'Beautify'. The 'Headers' tab has a count of 9. The 'Body' tab has a green dot icon. The 'JSON' dropdown is set to 'raw'. The 'Beautify' button is visible on the right.

History

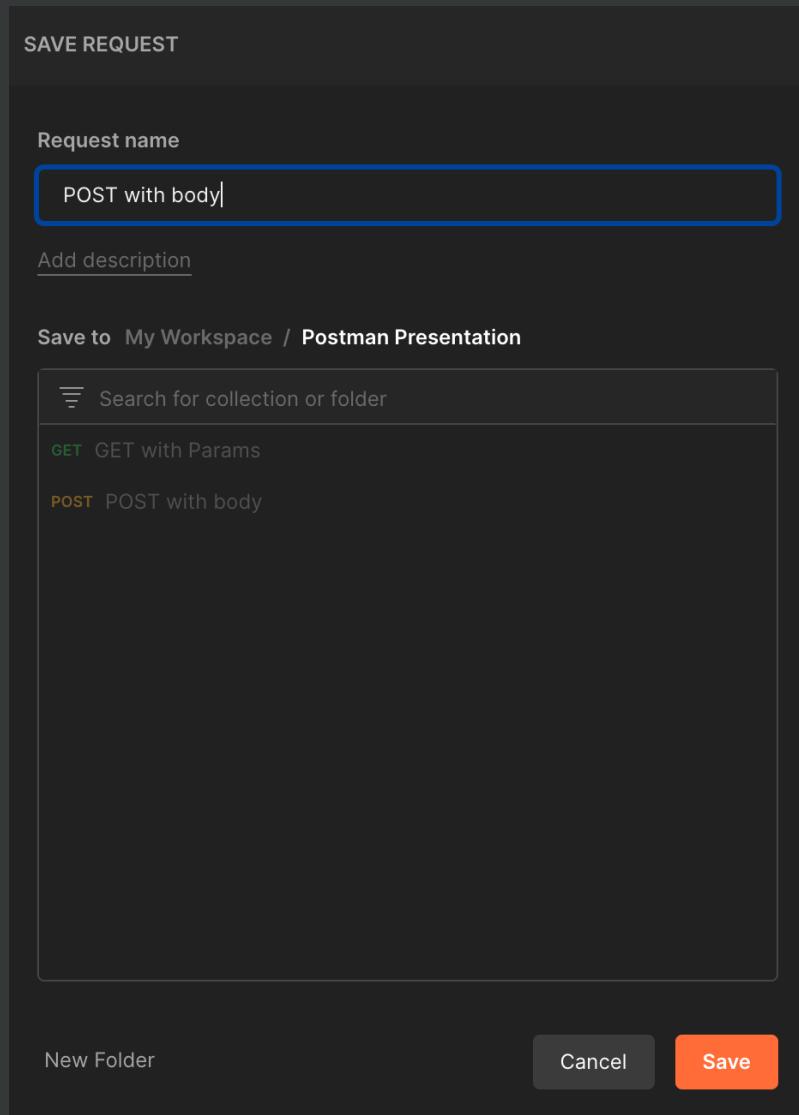
The screenshot shows the Postman application interface. On the left, there's a sidebar with tabs for Scratch Pad, Collections, APIs, Environments, Mock Servers, Monitors, and History. The History tab is currently selected. The main area displays a timeline of recent activity:

- Today:
 - POST https://httpbin.org/post
- Yesterday:
 - GET https://httpbin.org/get?query=search
 - GET https://httpbin.org/get?query=search
 - GET https://httpbin.org
 - GET https://httpbin.com
- June 23:
 - GET {{analytics_associate_service_uri}}/a
 - GET {{oauth_access_token_uri}}
 - GET {{analytics_associate_service_uri}}/a
 - GET {{adw_public_api}}/data-sets/25092

The main workspace shows a POST request to https://httpbin.org/post. The Headers tab shows 9 items. The Body tab is active, showing JSON input:

```
1 {  
2   "id": 1,  
3   "name": "Caleb"  
4 }
```

The response at the bottom indicates a 200 OK status with 324 ms and 852 B.



Save
a
Request
To
a
Collection

The screenshot shows the Postman collections view for a user named 'hy-vee'. It features a sidebar with 'Collections', 'APIs', 'Environments', 'History', and a folder icon. The main area displays a collection named 'Pharmacy API' containing various endpoints such as 'GraphQL', 'customers/{customerUuid}', 'stores/{storeCode}', 'orders/{orderNumber}', 'GET /stores/:storeCode/prescripti...', 'GET /stores/:storeCode/pickup-ti...', 'patients', 'invitations/{invitationId}', 'verified-identities', 'pickup-payments', 'DEL /approvals/:approvalId', 'POST /covid-adjustments', 'POST /primary-approvals', 'POST /child-approvals', 'GET /healthz', 'GET /rx-audits/:rxAuditId', 'POST This endpoint is called by McK...', and 'GET /version'.

Screenshot of the Postman application interface showing the "Pharmacy API" collection. The "Authorization" tab is selected, displaying configuration for JWT Bearer tokens.

Authorization Tab Configuration:

- Type: JWT Bearer
- Add JWT token to: Request Header
- Algorithm: RS256
- Private key: Select File (File path: {{jwt_private_key}})
- Payload (JSON):

```
{"iat": {{$timestamp}}, "nbf": {{$timestamp}}, "exp": {{expiration}}, "iss": "hy-vee-my-pharmacy"}
```

Collection Tree:

- Pharmacy API
 - GraphQL
 - customers/{customerUuid}
 - stores/{storeCode}
 - orders/{orderNumber}
 - GET /stores/:storeCode/prescri...
 - GET /stores/:storeCode/pickup...
 - patients
 - invitations/{invitationId}
 - verified-identities
 - pickup-payments
 - DEL /approvals/approvald
 - POST /covid-adjustments
 - POST /primary-approvals
 - POST /child-approvals
 - GET /healthz
 - GET /rx-audits/rxAuditId
 - POST This endpoint is called by M...
 - GET /version
 - POST /adult-invitations
 - GET /refill-details
 - POST /orders
 - POST /mail-order-payments
 - POST /mail-order-refunds
 - GET Search for agreement terms
 - POST /bulk-pcs-enrollments
 - POST /erp-enrollments
 - POST /prescription-transfers



The screenshot shows the Postman application interface with two prominent dialog boxes.

Import Dialog (Top Right):

- Header: "Import" with a close button "X".
- Tab navigation: File (selected), Folder, Link, Raw text, Code repository.
- Text area: "Drag and drop Postman data or any of the formats below".
- Format options: OpenAPI, RAML, GraphQL, cURL, WADL.
- Text: "OR"
- File upload: "Upload Files" button.

Export Collection Dialog (Bottom Left):

- Header: "EXPORT COLLECTION" with a close button "X".
- Text: "Postman Presentation will be exported as a JSON file. Export as:"
- Radio buttons:
 - Collection v2
 - Collection v2.1 (recommended) (selected)
- Text: "Learn more about collection formats" with a link.
- Buttons: "Cancel" and "Export" (orange).

The main workspace shows a JSON representation of a collection named "Postman Presentation" with one item, which includes a GET request to httpbin with parameters and headers.

GET ▼ https://api.dev.hy-vee.cloud/pharmacy/v1/stores/4f222734-65e4-4067-b399-99b6b14d5e1b/orders/50cc3c45-b60c-4507-92bd-6af24a291cd6?some_othe... Send ▾

Params ● Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

	Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/>	some_other_thing	[1,2,3,4,5,6]			
<input checked="" type="checkbox"/>	get_all	true			
	Key	Value	Description		

This is a lot to
copy and paste

The screenshot shows the Postman application interface. At the top, there's a navigation bar with links for History, Basics, Requests, Collections, Variables, Testing, and Insomnia. Below the navigation, a request configuration is displayed:

- Method: GET
- URL: `https://api.dev.ny-vee.cloud/pharmacy/v1/stores/4f222734-65e4-4067-b399-99b6b14d5e1b/orders/50cc3c45-b60c-4507-92bd-6af24a291cd6?`
- Params tab is selected.
- Body tab shows the query string: `some_other_thing=[1,2,3,4,5,6]&get_all=true`.
- Send button is visible on the right.

A modal window titled "Set as new variable" is open, overlaid on the main interface. It contains the following fields:

- Name: `pharmacy_api_uri`
- Value: `https://api.dev.ny-vee.cloud/ph`
- Scope dropdown: "Select a scope" with three options:
 - G Global (selected)
 - E Active Environment: Dev
 - C Collection: Pharmacy API

On the right side of the main interface, the text "Saving a Variable" is displayed in large white letters.

Environments

The screenshot shows the Postman interface with the 'Environments' tab selected. The 'Dev' environment is active, displaying a table of variables:

Variable	Type	Initial value	Current value
pharmacy_base_uri	default	https://api.dev.hy-vee.cloud	https://api.dev.hy-vee.cloud
jwt_issuer	default	https://dev-keycloak.hy-vee...	https://dev-keycloak.hy-vee.com/auth/re...
keycloak_uri	default	https://dev-accounts.hy-ve...	https://dev-accounts.hy-vee.com/auth/re...
jwt_private_key	secret	

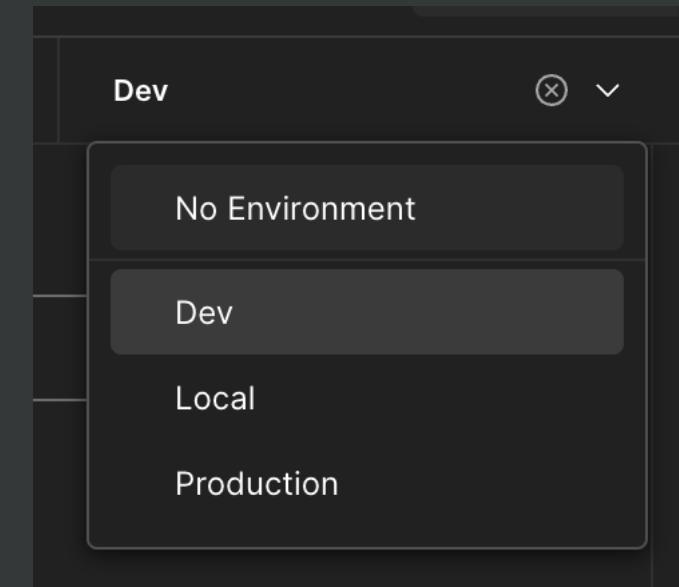
A modal at the bottom left provides instructions on using variables:

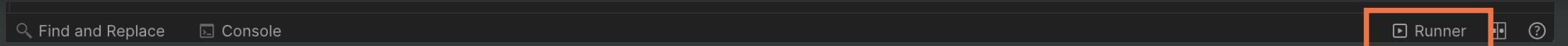
ⓘ Use variables to reuse values and protect sensitive data

Store sensitive data in variable type secret to keep its values masked on the screen. Learn more about [variable type](#)

Work with the current value of a variable to prevent sharing sensitive values with your team. Learn more about [variable values](#)

At the bottom, there are links for Online, Find and replace, Console, Runner, Capture requests, Cookies, Trash, and other navigation options.





Testing with the Runner

The screenshot shows the Insomnia Runner interface. At the top, there's a navigation bar with tabs for "Runner" (which is active), "Find and Replace", "Console", and "Runner" (which is highlighted with an orange border). Below the navigation is a large title "Testing with the Runner".

The main area is titled "RUN ORDER" and includes buttons for "Deselect All", "Select All", and "Reset". On the right side, there are configuration settings:

- "Iterations": A text input field containing "1".
- "Delay": A text input field containing "0" followed by "ms".
- "Data": A "Select File" button.
- A group of checkboxes:
 - Save responses (i)
 - Keep variable values (i)
 - Run collection without using stored cookies
 - Save cookies after collection run (i)
- A large blue "Start Run" button at the bottom right.

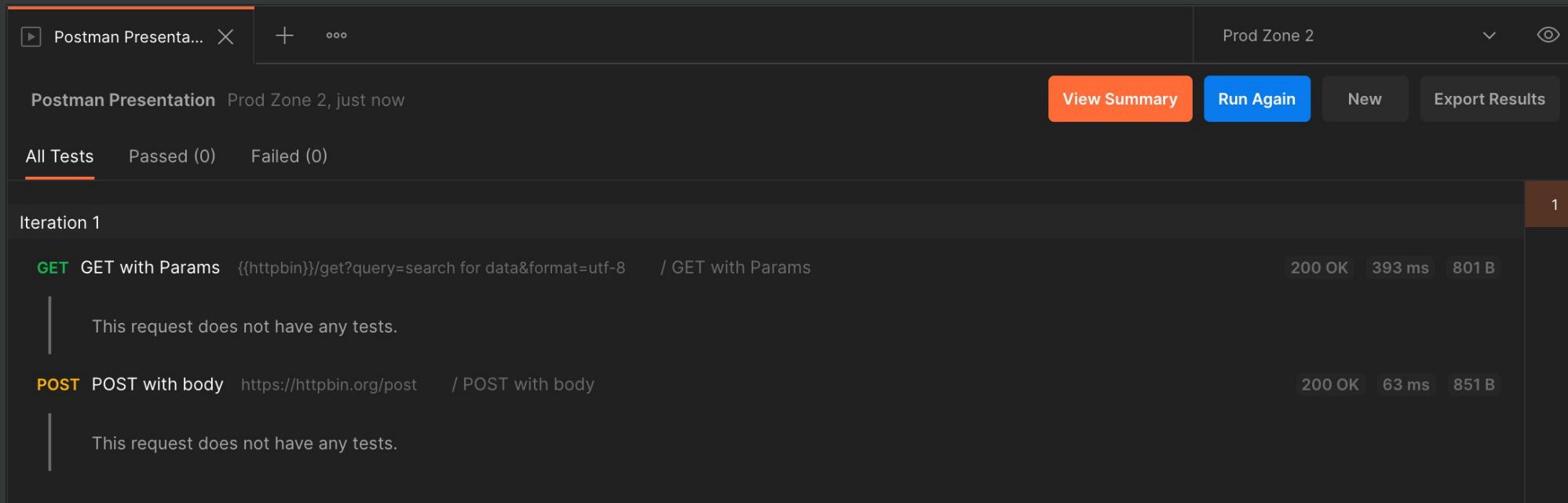
On the left side, there's a note: "Drag a collection or folder from your sidebar to get started". At the bottom left, it says: "You can find your previous collection runs in [history](#) or [import](#) past runs."

Add a Collection...

The screenshot shows the Postman Runner interface with the following configuration:

- RUN ORDER:** Contains two selected requests:
 - GET with Params
 - POST with body
- Deselect All | Select All | Reset** buttons are located above the run order list.
- Iterations:** Set to 1.
- Delay:** Set to 0 ms.
- Data:** A "Select File" button is available.
- Checkboxes (unchecked):**
 - Save responses
 - Run collection without using stored cookies
- Checkboxes (checked):**
 - Keep variable values
 - Save cookies after collection run
- Run Postman Presentation** button at the bottom.

...and press RUN



The screenshot shows the Postman application interface. At the top, there's a header with a play button icon, the title "Postman Presentation", a close button, and a "+" button. To the right of the title is the text "Prod Zone 2" with a dropdown arrow and an eye icon. Below the header, there are four buttons: "View Summary" (orange), "Run Again" (blue), "New" (grey), and "Export Results" (grey). Underneath these buttons, it says "Postman Presentation Prod Zone 2, just now". Below this, there are three tabs: "All Tests" (underlined in orange), "Passed (0)", and "Failed (0)". On the far right, there's a small brown box with the number "1". The main content area is titled "Iteration 1". It contains two requests: 1. A "GET" request titled "GET with Params" with the URL "{{httpbin}}/get?query=search for data&format=utf-8". It shows a status of "200 OK", a duration of "393 ms", and a size of "801 B". Below the request, a message says "This request does not have any tests." 2. A "POST" request titled "POST with body" with the URL "https://httpbin.org/post". It shows a status of "200 OK", a duration of "63 ms", and a size of "851 B". Below this request, a message says "This request does not have any tests."

There is no third step

```
~  npm i -g newman                                     Thu Jul  1 15:08:47 2021
npm WARN deprecated har-validator@5.1.5: this library is no longer supported
npm WARN deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.
npm WARN deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.
npm WARN deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.

changed 127 packages, and audited 128 packages in 6s

5 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
~ |
```

The logo for Aperture Laboratories, featuring a stylized camera lens on the left and the text "APERTURE LABORATORIES" on the right.

6.7s < Thu Jul 1 15:08:58 2021



```
● History • Basics • Requests • Collections • Variables • Testing • Insomnia •

newman run ./Postman\ Presentation.postman_collection.json
newman
Postman Presentation
→ GET with Params
GET https://httpbin.org/get?query=search for data&format=utf-8 [200 OK, 801B, 433ms]
→ POST with body
POST https://httpbin.org/post [200 OK, 851B, 73ms]

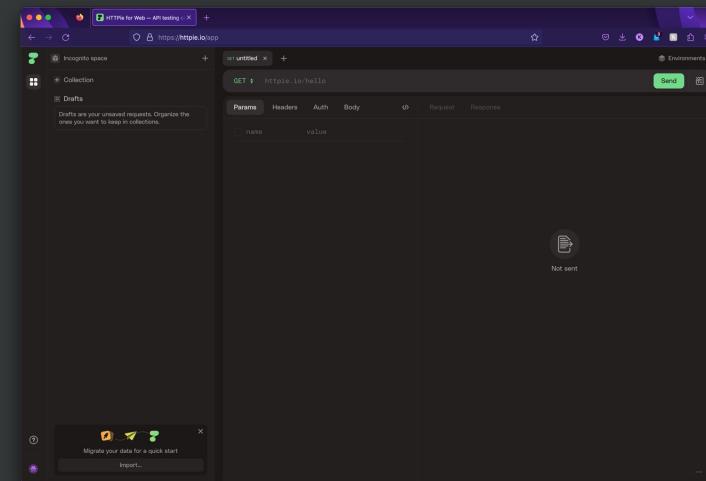
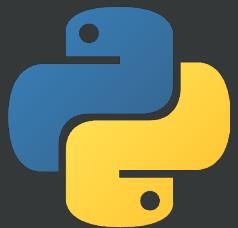


|                    | executed | failed |
|--------------------|----------|--------|
| iterations         | 1        | 0      |
| requests           | 2        | 0      |
| test-scripts       | 2        | 0      |
| prerequest-scripts | 2        | 0      |
| assertions         | 0        | 0      |


total run duration: 574ms
total data received: 1.16KB (approx)
average response time: 253ms [min: 73ms, max: 433ms, s.d.: 180ms]

1897ms < Thu Jul 1 15:15:08 2021
```

The Competition

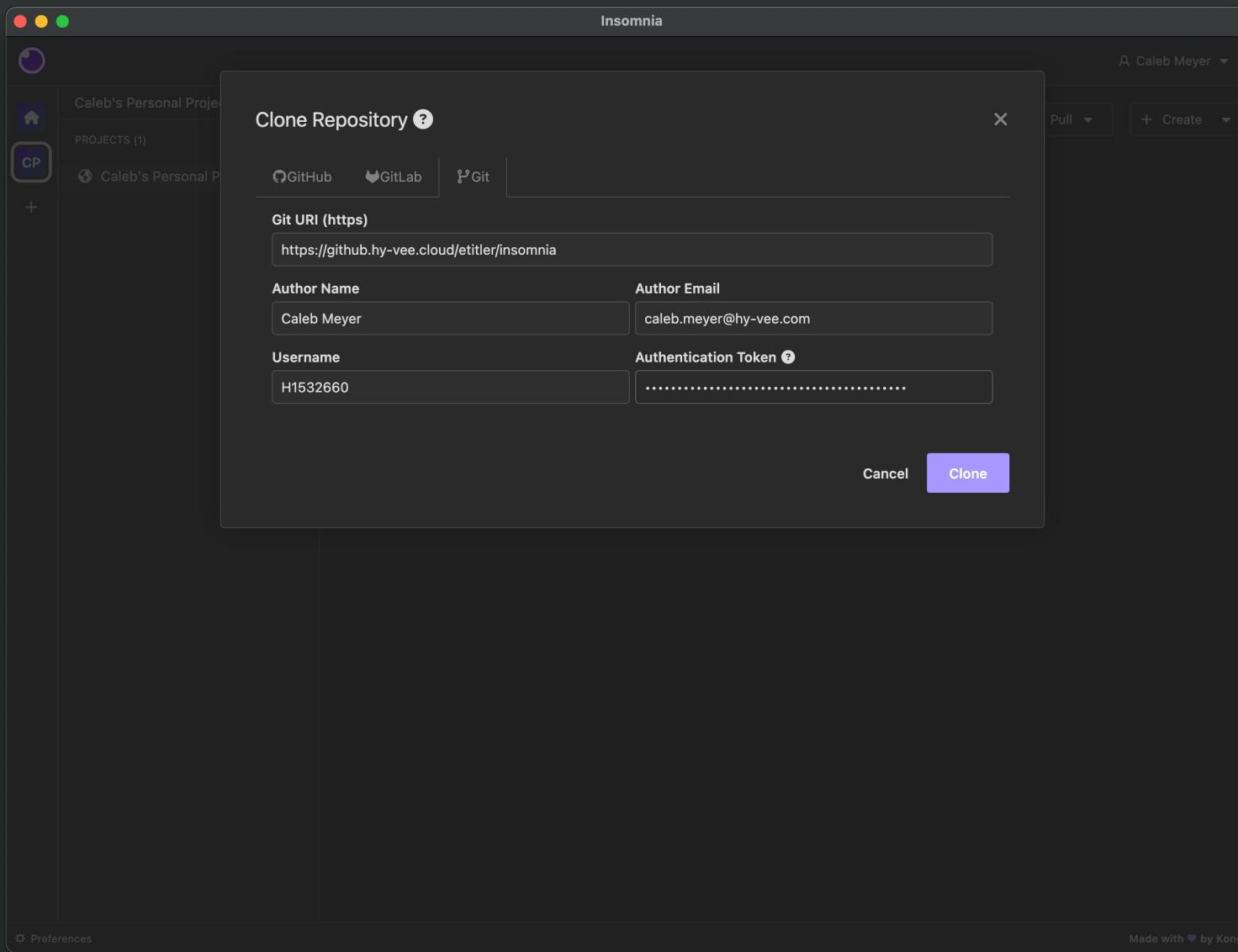


```
[~] http PUT httpbin.org/put hello=world
HTTP/1.1 200 OK
Access-Control-Allow-Credentials: true
Access-Control-Allow-Origin: *
Connection: keep-alive
Content-Length: 452
Content-Type: application/json
Date: Thu, 06 Dec 2018 15:43:38 GMT
Server: gunicorn/19.9.0
Via: 1.1 vegur
{
    "args": {},
    "data": "{\"hello\": \"world\"}",
    "files": {},
    "form": {}
```

httpie.io



Insomnia



Syncing is Easy!

• History • Basics • Requests • Collections • Variables • Testing • Insomnia •

Insomnia

Caleb's Personal Project / My Collection ▾ Setup Sync

A, Caleb Meyer ▾

No Environment ▾ Cookies

POST pharmacy_base_uri /graphql Send 200 OK 64.7 ms 823 B 8 Minutes Ago

GraphQL ▾ Auth ▾ Query Headers 1 Docs

getUnleashFeatures schema ↗

```

1 query getUnleashFeatures($featureNames: [String!]!, $properties: UnleashProperties) {
2   unleashFeatureToggles(featureNames: $featureNames, properties: $properties) {
3     active
4     featureName
5   }
6 }
```

schema not yet fetched

Query Variables ?

```

1 f
2 "featureNames": [
3   "my-pharmacy-delivery-and-pickup",
4   "my-pharmacy-download-smart-card",
5   "my-pharmacy-e-referral",
6   "my-pharmacy-combined-flu-and-covid-consent-flow",
7   "my-pharmacy-mail-order",
8   "my-pharmacy-pediatric-moderna",
9   "my-pharmacy-pill-reminder",
10  "my-pharmacy-prepay",
11  "my-pharmacy-schedule-in-mobile-app",
12  "my-pharmacy-vaccine-drive-thru"
13 ],
14 "properties": {}
15 }
```

Prettify GraphQL

\$.store.books[*].author

Preferences Made with ❤ by Kong

I'm not sleepy, I'm not sleepy, I'm not

• History • Basics • Requests • Collections • Variables • Testing • Insomnia •

Caleb's Personal Project / My Collection ▾ Set

No Environment ▾ Cookies POST pharmacy_base_uri /graphql Send ▾

Filter Bearer GraphQL Headers 1 Docs

Pharmacy API

GraphQL

Unleash Features

POST GQL Get Features

Vaccines

customers/{customerUuid}

stores/{storeCode}

patients

invitations/{invitationId}

verified-identities

pickup-payments

DEL /approvals/approvald

POST /covid-adjustments

POST /primary-approvals

GET /healthz

POST /child-approvals

GET /rx-audits/:rxAuditId

POST This endpoint is called by M...

GET /version

POST /adult-invitations

Pharmacy API + ⚙ Dev

Share Fork 0 Run Save ⚙

Pharmacy API

Overview Authorization Pre-request Script Tests Variables Runs

This authorization method will be used for every request in this collection. You can override this by specifying one in the request.

Type JWT Bearer

The authorization header will be automatically generated when you send the request. Learn more about [authorization](#) ↗

Add JWT token to Request Header

Algorithm RS256

Private key Select File

```
{{jwt_private_key}}
```

Payload

```
{"iat": {{$timestamp}}, "nbf": {{$timestamp}}, "exp": {{expiration}}, "iss": "hy-vee-my-pharmacy"}
```

The screenshot shows the Postman interface with a GET request. The URL is {{pharmacy_base_uri}}/stores/:storeCode/orders/:orderNumber. The 'Params' tab is selected, showing two path variables: storeCode with value 12345 and orderNumber with value blah. A 'Send' button is at the top right.

The screenshot shows the Insomnia interface with a GET request. The URL is pharmacy_base_uri/stores/:storeCode/orders/:orderNumber. The 'Query' tab is selected, showing storeCode=12345 and orderNumber=blah. A 'Send' button is at the top right.

Path Parameters

path parameters

Version 1.0.3 | Published 2 years ago

Overview

npm v1.0.3

This is a plugin for [Insomnia](#) that automatically replaces URL path parameters with syntax `:myParam` with values defined in the Query tab by the same name.

The screenshot shows the Insomnia application interface. The title bar reads "Insomnia". The top navigation bar includes "Caleb's Personal Project / My Collection", "Setup Sync", and "Caleb Meyer". On the left sidebar, there are icons for "Manage Environments", "Base Environment", "Sub Environments" (which is currently selected), and a "+" button. The main content area is titled "Base Environment" and displays the following JSON configuration:

```
1▼ {  
2  "pharmacy_base_uri": "https://api.dev.hy-vee.cloud",  
3  "laptop_password": "Hunter2"  
4 }
```

Below this, the text "Pls no hax" is displayed next to a yellow hands-in-prayer emoji.

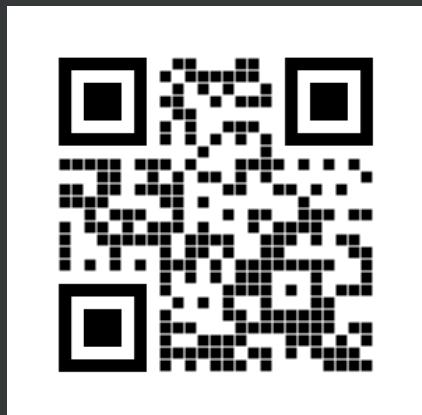
questions?

THANK YOU!

Links:

<https://github.hy-vee.cloud/pharmacy/pharmacy-api-postman>
<https://github.hy-vee.cloud/etitler/insomnia>

Slides can be found at
bit.ly/caleb-on-postman



Fonts used in this presentation:
Orbitron
Courier Prime
Ubuntu
Bradley Hand ITC
Merienda

Color Scheme (thanks colors.co):

- Background #343838 (cool gray)
- Foreground #FFFFFF (white)
- Accent 1 #005F6B
- Accent 2 #008C9E
- Accent 3 #00B4CC
- Accent 4 #00DFFC
- Accent 5 #E64823
- Accent 6 #EE7547 (Postman orange)