**Running Vets-api sample ping-pong webhook**

To help getting started utilizing the vets-api webhooks, we’ve provided a simple endpoint called “ping” and created a webhook event called ping-pong. After registering for the ping-pong event, you can send requests (ping) and then vets-api will respond within a minute (pong). This response from vets-api is an example of a webhook callback.

**Step 1: Installing Postman**

This guide uses Postman v8.11.1 which can be downloaded here:

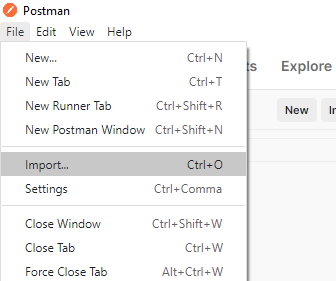
<https://www.postman.com/downloads/>

Once installed, create a Postman account and a workspace and familiarize yourself with the app. Introduction to Postman can be found here:

<https://learning.postman.com/docs/getting-started/introduction/>

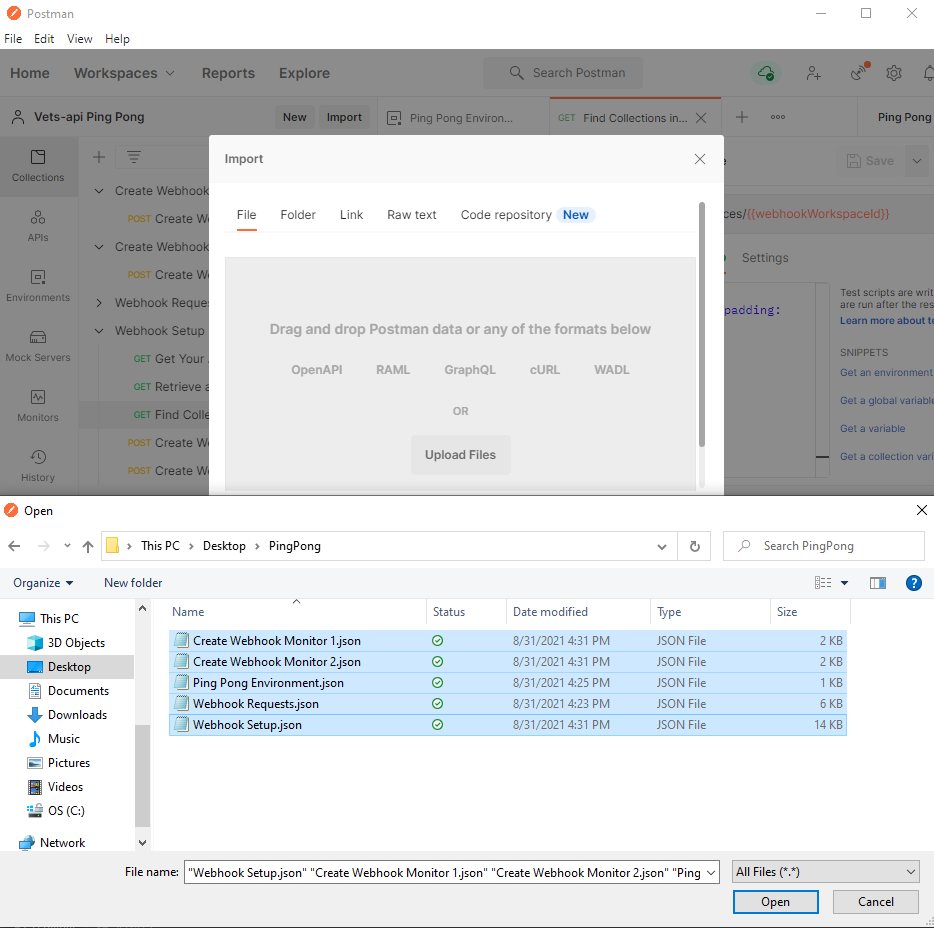
**Step 2:** **Importing the ping-pong project**

In Postman, select File->Import:



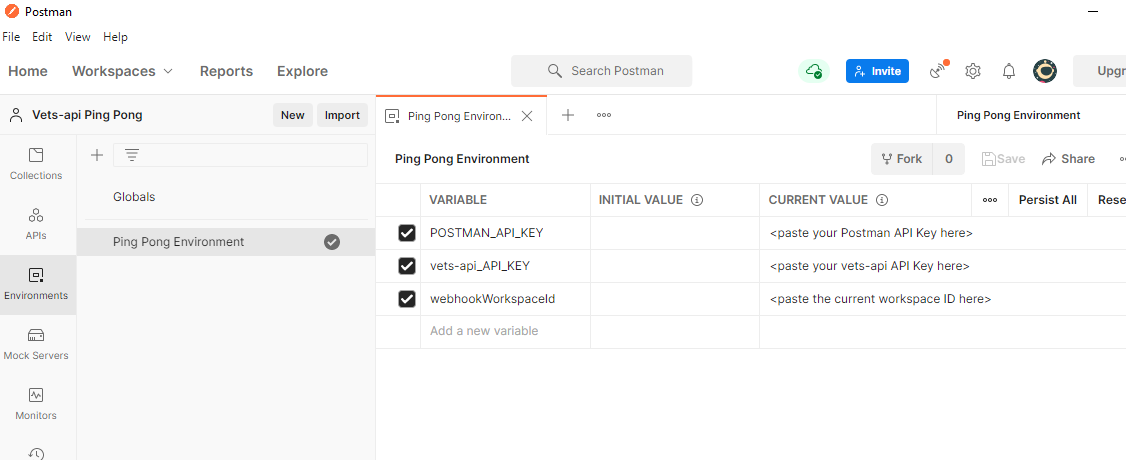
Import the following 5 files:

* Ping Pong Environment.json
* Create Webhook Monitor 1.json
* Create Webhook Monitor 2.json
* Webhook Requests.json
* Webhook Setup.json



**Step 3. Fill in Environment Variables**

Go to the environments tab to view the newly imported Ping Pong Environment. We need to fill in the first 2 values:

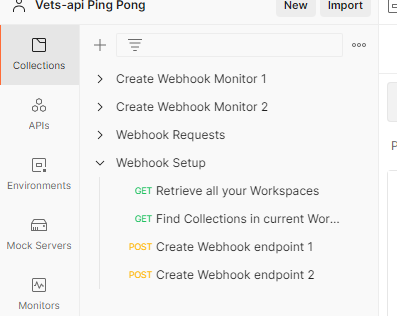


* POSTMAN\_API\_KEY: Visit <https://postman.co/settings/me/api-keys> to generate an API KEY and paste it in the CURRENT VALUE column of the environment.
* Vets-api\_API\_KEY: If you’ve been using vets-api previously, you likely already have an API Key, but a new one for non-production environments can be requested here: <https://developer.va.gov/apply>
* webhookWorkspaceId: Will be filled in in Step 4.

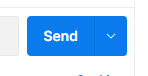
NOTE: Don’t forget to save changes to the environment with Ctrl-S.

**Step 4: Setting up Postman Monitors**

Navigate to the Collections tab and expand the Webhook Setup collection:



Click on the first request “Retrieve all your Workspaces”. Hit Send:



In the response body, copy the workspace ID for the workspace you are using. Below are my 4 workspaces, I am currently working in the “Vets-api Ping Pong” workspace, so I copied that workspace ID:

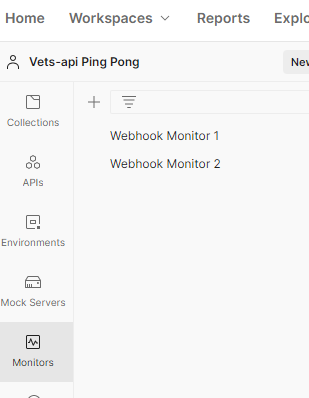
Graphical user interface, text, application, email

Description automatically generated

Paste this ID in the 3rd environment variable webhookWorkspaceID – don’t forget to save any environment changes. Next run the remaining 3 requests in the Webhook Setup collection.

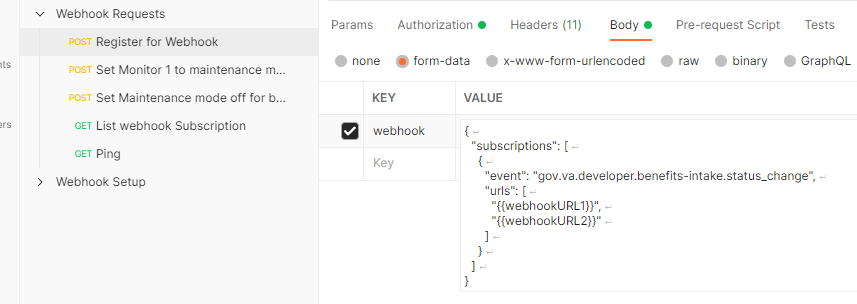
* “Find Collections in current Workspace”
* “Create Webhook endpoint 1”
* “Create Webhook endpoint 2”

You can verify setup was successful by checking the Monitors tab. You should see 2 Webhook Monitors. These Monitors will serve as HTTP servers that will be receiving the webhook responses and displaying them.



**Step 5: Registering for the Ping Pong webhook event**

Now we begin working with vets-api webhooks. To register for an event, we must send a POST request to the /register endpoint with a webhook object in the request body. You can see we are registering two URLs to the ping-pong event; these are the URLs for our Postman Monitors:

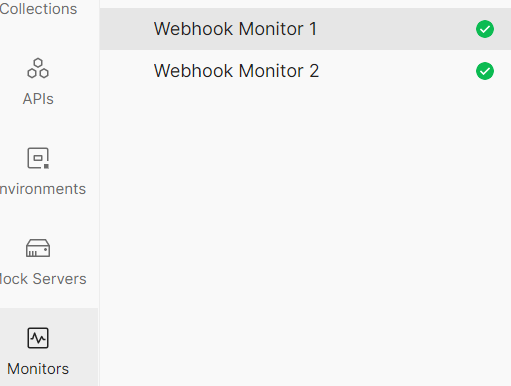


*webhookURL1 and webhookURL2 are the URL’s for the Postman Monitors created in Step 4.*

Send the “Register for Webhook” request to register for the ping-pong event.

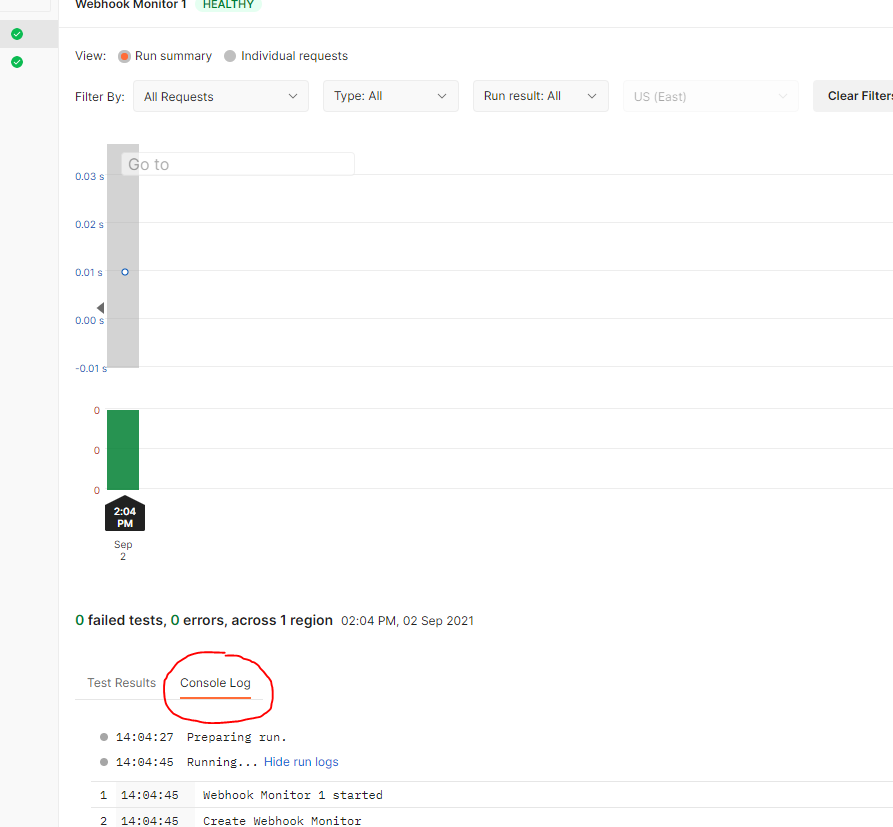
**Step 6:** **Sending Ping request and receiving Pong response**

Send the “Ping” request and wait a minute. You should see the callback response in the Monitors tab:



**Step 7: Viewing the response data**

Click on a Monitor. Each response sent to the Monitors will show as a green bar in the Monitor chart. Click on the green bar and select the Console Log tab:

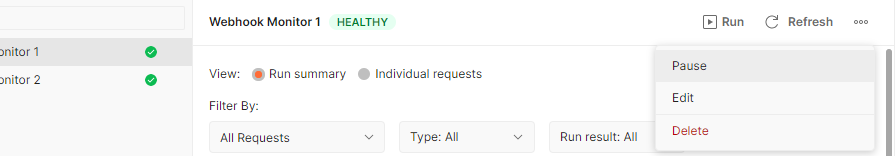


Scroll down to see the data object. The notifications array displays the pong message.

**Robustness through subscriber downtime**

We have provided solutions if a subscriber server is unreachable.

1. Retry attempts: If a message sent to a callback URL fails, we retry the message after a short period of time. You can demonstrate this to yourself by pausing one of the Postman monitors by click the 3 dots in the top right and selecting pause:



Send another Ping request or two. Wait a couple of minutes, then unpause the Monitor. You will see the Monitor receive responses shortly. It is not feasible for us to retry endlessly, so at some number of retries, we no longer attempt to send requests to a downed URL. The number of retries is dependent on the event registered for.

1. Maintenance Alerting: If you know a callback URL will be unreachable for a period, you may declare that URL in maintenance mode. In the Webhook Requests Postman Collection, we have provided requests to demonstrate the maintenance requesting. Run “Set Monitor 1 to maintenance mode” and run Ping requests. Notice how only Monitor 2 is receiving responses now. We have also provided the “Set Maintenance mode off for both” request to re-enable sending to both Monitors.

**Viewing webhook subscriptions**

You may want to see what URLs you have subscribed to which vets-api events. We have provided a ‘list’ endpoint to display each subscription. Run the “List webhook Subscription”. Since we’ve only registered for the ping-pong event in this guide, that is the only subscription displayed.