

Using Postman to query the Open-Audit API

I often utilise Postman to query the Open-Audit API when developing.

Just using a browser, it's difficult to send anything other than a GET request - but Postman makes it simple to send a POST, PATCH or DELETE as required.

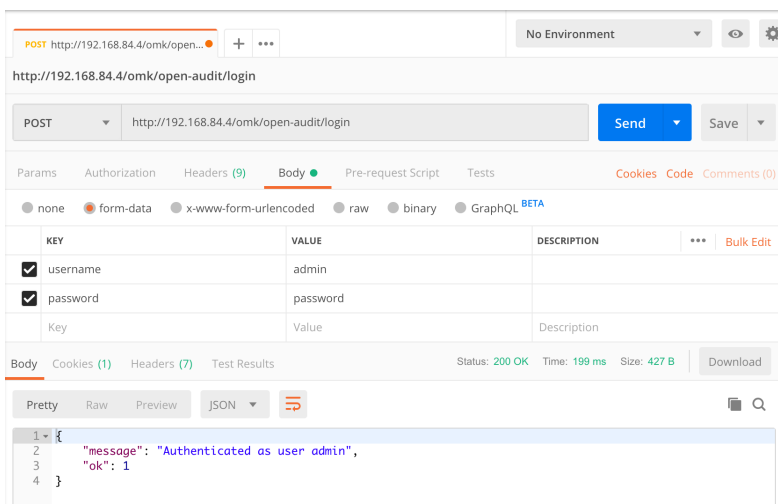
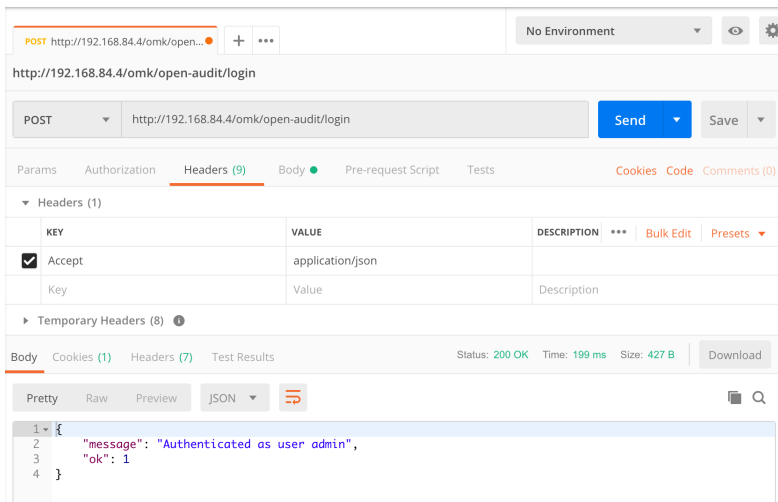
You can get it from <https://www.getpostman.com/downloads/> for Windows, Mac and Linux.

Install and start Postman. You can elect to create an account or not. You can also elect to create a new item using the wizard, or just close the modal and jump in. Let's do that 😊

For the below, my Open-Audit server is running on 192.168.84.4. You should substitute the IP address of your Open-Audit server.

Logon

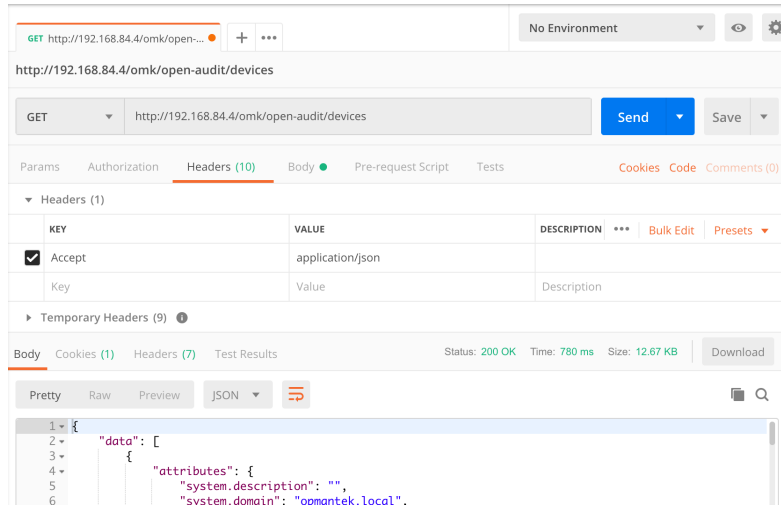
First you need to make a post to /login to get a cookie. Set the dropdown to POST and the URL to ***http://192.168.84.4/omk/open-audit/login***. Set the header Accept to application/json. Set the Body to form-data and provide the username and password keys, with values as appropriate for your installation. By default it will look as below. Now click the Send button.



You should see the JSON result saying you have been authenticated.

Read

Once that's done, it's time to request some data. Make a GET request to ***http://192.168.84.4/omk/open-audit/devices*** and you should get a JSON response containing a list of devices. You can see the start of the JSON in the screenshot below.



What about changing the attribute of an item? Not too difficult. You'll need the ID of the device you want to change, along with the attribute name from the database and an access token.

Access tokens are generated with every request type. You will need a token when submitting a POST or PATCH request. Run another query (a GET is fine, even if no items are returned) and parse the JSON response for meta.access_token. Include this in your request body as below.

Attribute names are visible in the application by going to menu Admin Database List Tables and clicking on the "system" table. Let's change the description for our device with ID 14.

You'll need to create a JSON object and assign it to the "data" item to do this. It's not too difficult. Your JSON object should look like below (formatted and indented for easy reading).

```
{
  "data": {
    "access_token": "bbc0c85653fdc4b83d108cba7641bfcbbbc77586dfb8f32d08973770a90fe",
    "id": "14",
    "type": "devices",
    "attributes": {
      "description": "My New Description"
    }
  }
}
```

It looks worse than it is. Normally you would use code to do this, so it's a simple two line conversion. Because we're using Postman, we'll have to do it ourselves. A useful site is <https://jsonlint.com/>

So now you have your payload, let's send it to Open-Audit. Make a new PATCH request and use the URL ***http://192.168.84.4/omk/open-audit/devices/14***. Supply the data attribute in the body x-www-form-urlencoded section and hit Send. You should see the request as below.

PATCH http://192.168.84.4/omk/open-audit/devices/14

Send Save

Params Authorization Headers (10) **Body** Pre-request Script Tests Cookies Code Comments (0)

none form-data x-www-form-urlencoded raw binary GraphQL BETA

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> data	["data":{"id":"14","type":"devices","attribut...	
Key	Value	Description

Body Cookies (1) Headers (7) Test Results Status: 200 OK Time: 524 ms Size: 7.29 KB Download

Pretty Raw Preview JSON

```

15  "contact_name": "",
16  "credentials": "[12]",
17  "dbus_identifier": "",
18  "description": "My New Description",
19  "discovery_id": 6,
20  "dns_domain": "",
21  "dns_hostname": "",
  
```

Delete

Deleting an item is the even easier. Let's delete an Org. In this case, our Org with ID 2. Make a new DELETE request to ***http://192.168.84.4/omk/open-audit/orgs/2***. That's it - easy 😊

And if we want to read a specific entry, it's just a GET request. Let's get our default Org - ID 1. Just make a GET to ***http://192.168.84.4/omk/open-audit/orgs/1***.

Execute

What about running a query? What's the HTTP verb used to EXECUTE something? There is none 😞 But we'll make do by supplying /execute after the ID. So to execute a query, make a GET request to ***http://192.168.84.4/omk/open-audit/queries/1/execute***. To execute a discovery, task or baseline, use the same format - ID/execute.

Create

If we want to create a new item, use a http verb of POST^ and include an access token. An access token is generated with every request (except logon) and any of the last 20 (by default, settable in the configuration) will be accepted. You should always aim to use the last access token issued.

Remember we always receive the result in JSON as that is in our request header. We could receive it as HTML is we want - just remove that header item. Maybe more useful is a CSV output. Remove the Accept header and change the URL for a GET to ***http://192.168.84.4/omk/open-audit/queries/1/execute?format=csv***. Done - CSV output you can copy and paste into Excel.

It really is that simple. The only one to watch is the PATCH request, because you have to create your own JSON. Just about everything else is quite discoverable. Make sure you check the pages for [Collections](#) which detail the request formats. And don't forget [The Open-Audit API](#) page as well.

That makes for a simple and easy way to test the Open-Audit API.

For more examples, please our new page [API Examples for Postman](#)

Onwards and upwards.

Mark Unwin.