

The Open-Audit API (1.12.8)

- [Introduction](#)
- [Open-Audit's API](#)
- [Options](#)
 - [Format](#)
 - [Sort](#)
 - [Current](#)
 - [GroupBy](#)
 - [Limit](#)
 - [Offset](#)
 - [Properties](#)
 - [Filter](#)
 - [Include](#)
 - [Version](#)
 - [Action](#)
- [Routing Table](#)
- [Devices](#)
 - [Examples of retrieving data](#)
 - [Example of updating a device](#)
 - [Device sub_resource Names](#)

Introduction

Open-Audit is implementing a JSON Restful API. The format of the request URLs will also be used in the web application (with a few extra pieces, see below re: action).

In short, you can call /devices using the API to retrieve a JSON document containing the devices as per the JSONapi.org spec.

You can also call /devices in the web front end and retrieve the same list which is displayed in the web page and formatted as per the normal application theme.

Filtering, sorting and the other options apply equally to both the web frontend and to the JSON API.

NOTE - This API is not ready for a full release as yet and items below are subject to change. As at 1.12.8, this is how it stands.

NOTE - This page is incomplete and is being updated as we work towards a released version of the API.

Open-Audit's API

Open-Audit is basing its API on <http://jsonapi.org> with the intention of providing simple and intuitive access in a manner familiar to developers. In addition to this API, the web interface will use the same request/URI format and supply some additional actions.

Options

Format

Using the format option is useful when using a web browser but you wish to see the result in JSON format. Adding format=json achieves this. Normally a web browser will set its accept header to html, so in that case we return the rendered page. Using an API to retrieve JSON you should set the accept header to contain the string "json". That might be "json/application" or whatever you like. You can override this by providing the format option in the URL..

We tend to use the Google Chrome extension called Postman for testing actual restful queries. You might like to install and test with that. <http://www.getpostman.com>.

```
format={json}
```

Sort

To sort by a database column, use "sort={attribute}". To reverse sort, insert a minus, thus "sort=-{attribute)".

```
sort=[-]{attribute}
```

Current

By default, only attributes with "current=y" are retrieved. To override this, set current as below.

```
current={y|n|all}
```

GroupBy

```
groupby={attribute}
```

Limit

When requesting JSON, by default no limit is set. When requesting screen display, limit is set to 1000 by default.

```
limit={int}
```

Offset

The offset is the count of devices you wish to return data from.

```
offset={int}
```

Properties

Requested properties should be in a comma separated list.

```
properties={attribute 1},{attribute 2},{attribute 3}
```

Filter

To filter by a property value, use the property name. Operators that should precede the value are !=, >, >=, <, <=, 'like' and '!like'. If no operator is specified, the default is =.

```
{attribute}=[operator]{value}
```

Include

When requesting the details for a resource (a device), if the request is JSON based only the 'system' table will be returned. NOT all the related tables. There will be links for these tables in the data->links section. If you would like the system table AND the bios table (for example) in a JSON request, you can use the 'include' keyword. An example would be a GET request with accept header as JSON to <http://server/open-audit/index.php/devices?include=bios>. For a HTML (web browser) formatted request, by default all related tables are included so the web page can render the device details as it has always done. When using include, you can use the keyword 'all' to retrieve all tables that contain related information.

```
include={sub_resource}
```

Version

To request a different version of the API (currently only v1 exists), use the url `/api/{version}/devices` or `/v1/devices`.

Action

When using the API the default action is determined according to the request method and URL. You can override this by providing the 'action' option in the URL. This is only really required when using the web front end. Normal use of the API does not require you to set 'action'. An example of this is when creating a new item. You would normally use POST to /item but in the case of a web user, you need a web form to be able to fill out the item details. In that case there is no facility for this in a typical JSON restful API. We work around this by providing `action=create` in a GET request for the URL. IE - <http://server/open-audit/index.php/networks?action=create>.

Routing Table

The default action if nothing matches below is to return a collection of items.

* Not all routes are available on or apply to all endpoints.

** Actions ending in `_form` are for web form input

Request Method	id	action	sub_resource	sub_resource_id	Resulting Action	Notes	URL Example
GET	N				collection		/devices
GET	N	create			create_form		/devices?action=create
GET	Y/N	create	Y		sub_resource_create_form		/devices?action=create&sub_resource=credentials
GET	N	import			import_form		/devices?action=import
GET	Y				read		/devices/{id}
GET	Y	update			update_form		/devices/{id}?action=update
GET	Y	download			download		/scripts/{id}?action=download
GET	N	update			bulk_update_form	Should provide attribute "ids" which is a comma separated list of ID's upon which to bulk update.	/devices?action=update&ids=1,2,3,4
POST	N				create		/devices
POST	N	import			import		/devices?action=import
POST	N	edit			bulk_update_form		
POST / PUT / PATCH	Y				update		/devices/{id}?action=update
POST	Y		Y		sub_resource_create		/devices/{id}?sub_resource=credential
DELETE	Y		N		delete	You cannot delete a default org, location or script.	/scripts/{id}
DELETE	Y		Y	Y	sub_resource_delete		/devices/{id}/credential/{sub_resource_id}
ALL		<empty> or list			collection		

Devices

Examples of retrieving data

HTTP Verb	Accept	URL	Result	Example Response
GET	JSON	/devices	Get a list of all devices.	devices.json
GET	JSON	/devices?properties=system.id,system.name,system.type,system.serial	Get a list of devices with the data id, name, type and serial	devices_properties.json
GET	JSON	/devices?system.os_group=Windows	Get a list of devices with an OS Group of Linux.	devices_os_group_linux.json
GET	JSON	/devices?sub_resource=ip&ip.network=192.168.1.0/24&properties=system.id,system.name,system.domain,ip.ip	Get a list of devices in the 192.168.1.0/24 subnet.	devices_in_192-168-1-0.json
GET	JSON	/devices/8	Get the system table of device with id = 8.	devices_8.json
GET	JSON	/devices/8?include=bios	Get the system table and bios table for device with ID = 8.	devices_8_include_bios.json
GET	JSON	/devices/8/bios	Get the bios details (not the system table) for device with ID = 8.	devices_8_bios.json

Example of updating a device

HTTP Verb	Accept	URL	Data	Result	Example Response
PATCH	JSON	/devices/8	<pre>{ "data": { "type" : "devices", "id" : 8, "attributes" : { "description" : "my HP" } } }</pre>	Update the description of the device with ID = 8.	devices_8_patch_description.json

Device sub_resource Names

NAME	NAME	NAME
audit_log	netstat	service
bios	network	share
change_log	optical	software
credentials	pagefile	software_key
disk	partition	sound
dns	print_queue	task
edit_log	processor	user
ip	route	user_group
log	san	variable
memory	scsi	video
module	server	vm
monitor	server_item	windows
motherboard		