

# An exploration of NODE-RED and API calls towards the Meraki Dashboard

## 1. Getting node-red installed

This seems not as easy as just running an 'apt-get' on Linux.. the installation consists of two parts: Node.js which is a JavaScript program where node-red should be running on top of, and the node called "node-red".

For this exploration I installed Node.js with "\$ brew install node" on OSX.

This contains the "node" environment. With the node package manager "npm" we should be able to now install node-red:

```
sudo npm install -g --unsafe-perm node-red
```

Credits due: I stole the installation information from:

<http://developer.opto22.com/nodered/general/getting-started/node-red-install-red/>

## 2. Starting node-red

Start the node and observe the output:

```
$ node-red -v
11 Dec 09:57:35 - [info]

Welcome to Node-RED
=====

11 Dec 09:57:35 - [info] Node-RED version: v1.2.6
11 Dec 09:57:35 - [info] Node.js version: v15.4.0
11 Dec 09:57:35 - [info] Darwin 19.6.0 x64 LE
[...]
11 Dec 09:57:35 - [info] Server now running at http://127.0.0.1:1880/
11 Dec 09:57:35 - [info] Starting flows
11 Dec 09:57:35 - [info] Started flows
```

You should now be able to browse to node-red on <http://127.0.0.1:1880/>.

## 3. Expanding with meraki-nodes

According to Devnet, you now can install pre-cooked Meraki nodes:

<https://developer.cisco.com/meraki/meraki-dashboard-api-node-red-node-#!/getting-started/getting-started>

So :

```
$ cd .node-red/
$ npm install node-red-contrib-meraki-dashboard-api
```

This might not go as planned.. as is common with opensource stuff.

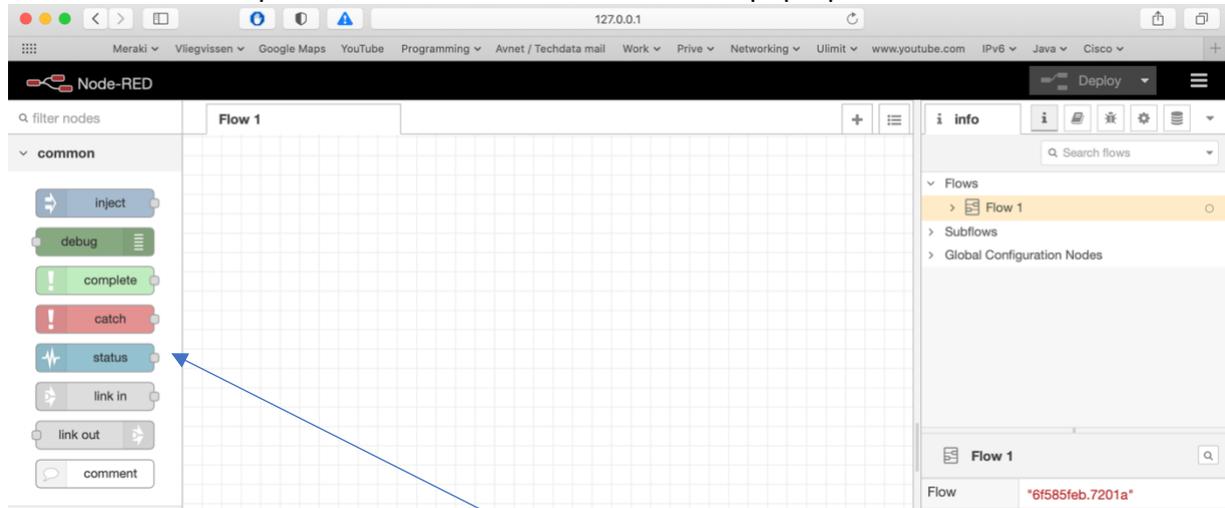
Now **stop-and restart the node-red program**. Afterwards browse to the webpage.

# An exploration of NODE-RED and API calls towards the Meraki Dashboard

## 4. Configuring our first input and API call

The surprising thing is that IBM (!) came up with node-red and it was initially used for hardware development and IoT. (?!)

Browse to the URL provided above and node-red should pop-up:



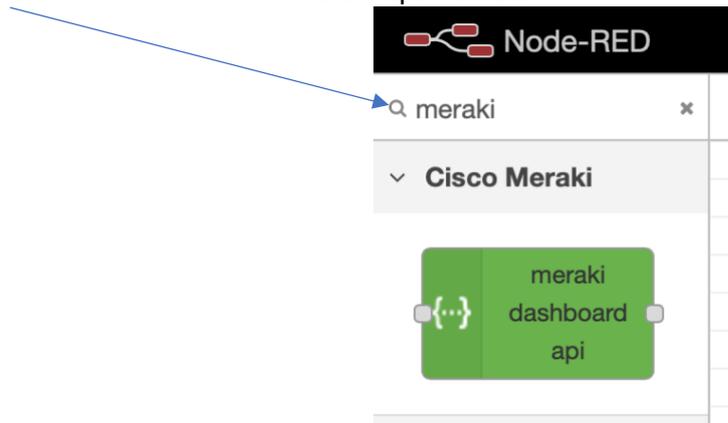
The pretty colored boxes on the left are “nodes” and they are part of your “Palette”.

## 5. Making the first node-red flow

Credits due once more: <https://www.youtube.com/watch?v=9PffwMa ITE>

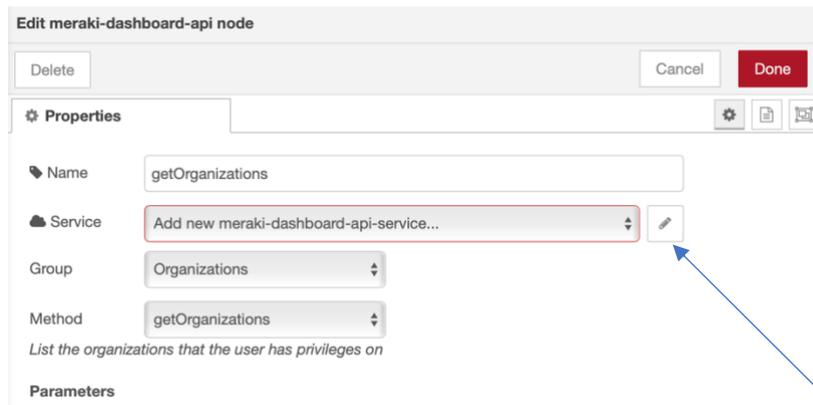
Here we are going to make a Flow within node-red from left-to-right. A flow is an event that runs through a collection of connected nodes.

Search for the meraki dashboard api – node:

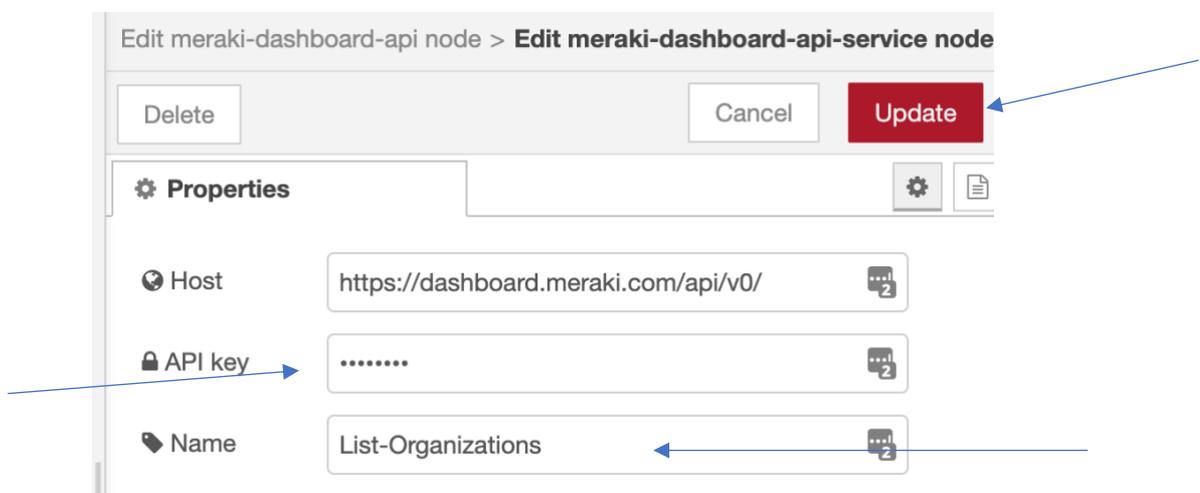


Pull this node from your Palette to the canvas and configure it by Double-Clicking:

## An exploration of NODE-RED and API calls towards the Meraki Dashboard



Enter a Name, Select the Group (Organizations) and Method (getOrganizations). Add the API key for the calls and provide a relevant name.



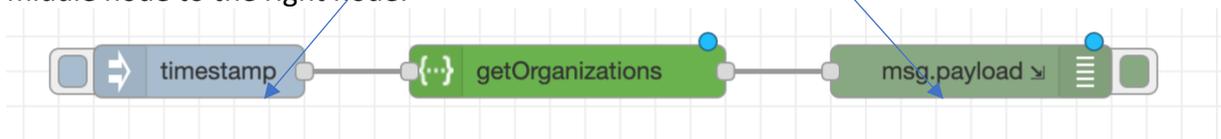
Once you have updated these properties, click **Update**. Click **Done** within the former frame.

For Input (Common section): We'll select a grey "Inject" node and drag it to the canvas.

For Output (Common section): We'll select a green Debug node and drag it to the canvas on the right.

The Inject node will trigger an event (eg an API call) and the Debug node returns the result.

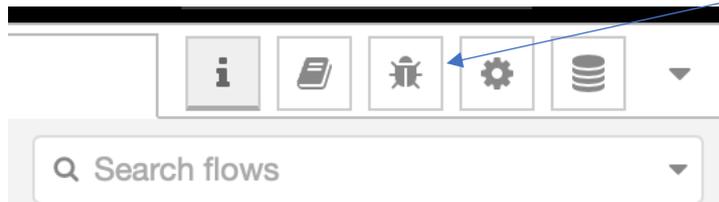
Click on the end of the node on the left, and connect it to the middle node. Connect the middle node to the right node:



### 6. Generating the call

On the right side of the page make sure you select the "debug output" (bug symbol):

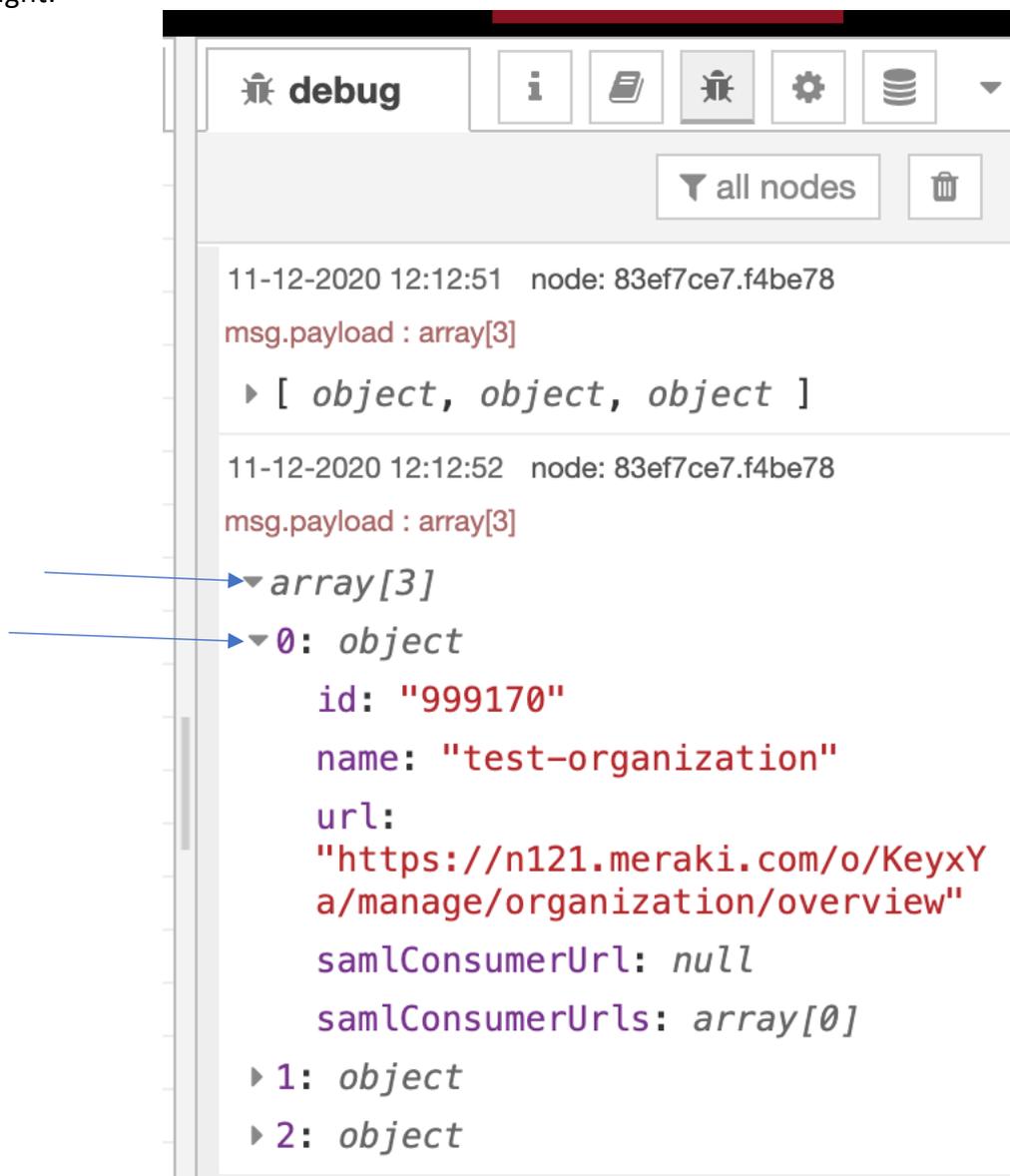
## An exploration of NODE-RED and API calls towards the Meraki Dashboard



To trigger the call, click on the inject node where it says "timestamp":



This should trigger the call. The output can be observed in the "Debug" Windows on the right:



This is your first API call to the Meraki Dashboard, without ANY programming...

## **An exploration of NODE-RED and API calls towards the Meraki Dashboard**

Note: By importing other pre-cooked flows within node-red, you have complete flows available to create networks, users, SSIDs.. etc.

**“Enjoy!”**