Integration with NETIO PowerPDU

Document number: PO-182-EN Version: 1.0 Date of publication: October 6, 2022

Introduction

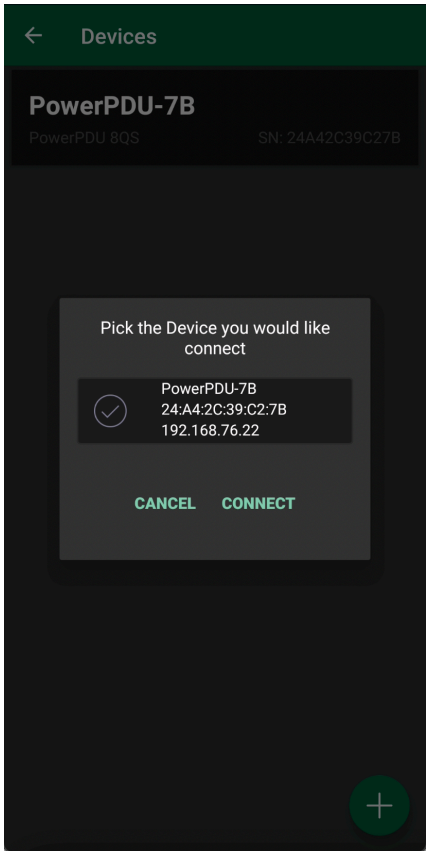
Ampio enables integrations via the Node-RED platform and makes a library of nodes available, which makes different types of integrations easier. The basics of integrations with the Ampio system carried out via Node RED are described in the Integration of the Ampio system with Node-RED manual.

This guide will take you through the steps of integrating your Ampio installation with Netio PowerPDU. For the purposes of this document, the NETIO PowerPDU 8QS model was used.

Connecting the PDU

The first step is to connect your Netio PowerPDU to power and an Ethernet cable from your local network.

Then, you will need to find out the device's IP address. In order to do that, download the NETIO Mobile application and connect your phone to the same network as the NETIO unit. Go to the *Devices* option in the app's menu and click on *Find device*. The app will return information about your connected NETIO device, including the IP address.



Save the IP address to use it later in Node-RED.

Netio.Json

The NETIO PowerPDU provides various measuring tools for each of its outputs. For the purpose of integrating this functionality with your Ampio installation, you will need access to your Netio JSON configuration. It can be retrieved by pasting the following URL in your browser **http://*device-IP*/netio.json**. What you should see is presented in the figure below.


```
{
"Agent":{"Model":"8QS","DeviceName":"PowerPDU-7B","MAC":"24:A4:2C:39:C2:7B","SerialNumber":"24A42C39C27B","JSONVer":"2.4","Time":"2022-07-
19T08:57:40+01:00","Uptime":65178,"Version":"3.2.2","OemID":600,"VendorID":0,"NumOutputs":8,"NumInputs":1},
"GlobalMeasure":
{"Voltage":238.33,"TotalCurrent":0,"OverallPowerFactor":1.00,"TotalPowerFactor":1.00,"OverallPhase":0,"TotalPhase":0,"Frequency":50.05,"TotalEnergy":0,"TotalReverseEnergy":0,"TotalEnergyNR":0,"TotalReverseEnergyN
R":0,"TotalLoad":0,"EnergyStart":"1970-01-01T00:00:00+01:00"},
"Outputs":[
{"ID":1,"Name":"Power output 1","State":0,"Action":6,"Delay":5000,"Current":0,"PowerFactor":1.00,"Phase":0.00,"Energy":0,"ReverseEnergy":0,"EnergyNR":0,"ReverseEnergyNR":0,"Load":0},
{"ID":2,"Name":"Power output 2","State":0,"Action":6,"Delay":5000},
{"ID":3,"Name":"Power output 3","State":0,"Action":6,"Delay":5000},
{"ID":4,"Name":"Power output 4","State":0,"Action":6,"Delay":5000},
{"ID":5,"Name":"Power output 5","State":0,"Action":6,"Delay":5000},
{"ID":6,"Name":"Power output 6","State":0,"Action":6,"Delay":5000},
{"ID":7,"Name":"Power output 7","State":0,"Action":6,"Delay":5000},
{"ID":8,"Name":"Power output 8","State":0,"Action":6,"Delay":5000}
],
"Inputs":[
{"ID":1,"Name":"Input 1","State":0,"S0Counter":0}
],
"PAB":[
{"Type":"RANGE","Name":"PAB_RANGE_1","Enabled":false,"In":false},
{"Type":"ZONES","Name":"PAB_ZONES_1","Enabled":false,"Zone":0}
],
"Watchdogs":[
{"Name":"default_pinger","Enabled":false,"Fail":false,"LastStatus":false,"Timestamp":0}
],
"Rules":[
{"Name":"Rule1","Enabled":false,"Result":false},
{"Name":"Rule2","Enabled":false,"Result":false},
{"Name":"Rule3","Enabled":false,"Result":false}
]}
```


More useful information on JSON structure in the context of NETIO products can be found on the PDU manufacturer's website

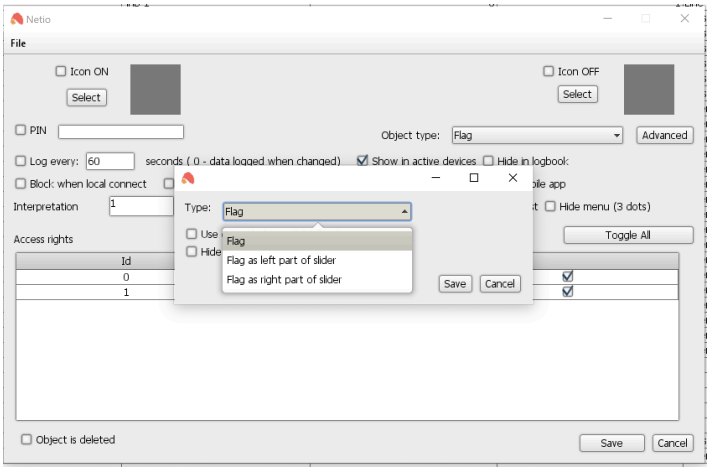
Adding objects to Smart Home Manager

In order to integrate NETIO PowerPDU with Ampio you must carry out two separate configurations in the Smart Home Manager, i.e. one for obtaining status from the NETIO device and another one for the controlling device.

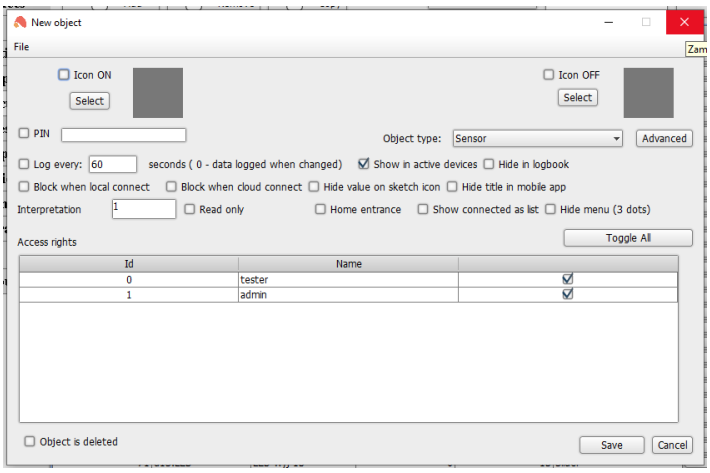
First, open your Smart Home Manager and navigate to "Objects" on the left side menu. Then, in the top bar, click on the "Add" button to add new objects. The number of objects you create here should correspond to the number of NETIO outputs that you wish to use, plus any values, e.g. voltage or current that you wish to measure. It might be helpful to change the Description of created objects to e.g. *Netio output 1*, *Netio output 2*, *Voltage*, etc.

1403	200:	Voltage	0	1	32 bit Value	Settings
1407	200:	Current	0	2	32 bit Value	Settings
1408	cef6:	Output 1	0	1	Flag	Settings
1409	cef6:	Output 2	0	2	Flag	Settings
1410	cef6:	Output 3	0	3	Flag	Settings
1411	cef6:	Output 4	0	4	Flag	Settings
1412	cef6:	Output 5	0	5	Flag	Settings
1413	cef6:	Output 6	0	6	Flag	Settings
1414	cef6:	Output 7	0	7	Flag	Settings
1415	cef6:	Output 8	0	8	Flag	Settings
1416	200:	Energy	0	9	32 bit Value	Settings
1417	200:	Frequency	0	10	32 bit Value	Settings

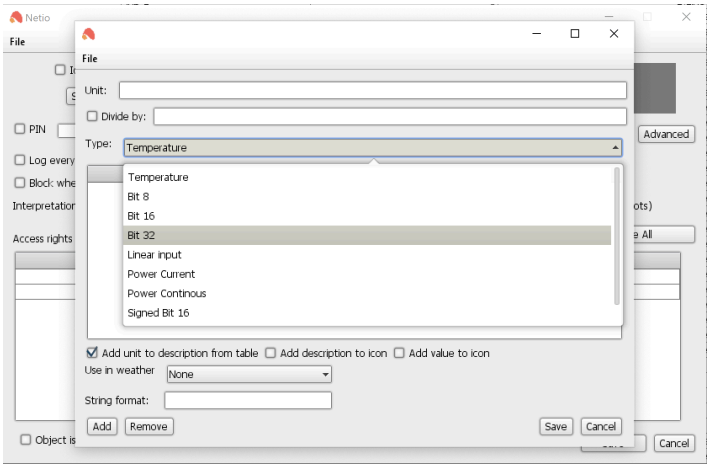
Next, go to each of the output objects' settings and set their object type to "Flag".



In a similar fashion, for the objects created to perform measurements, select the type “Sensor”, as shown below.



Then, click the “Advanced” button and in the new pop-up window navigate to “Type” (which is, by default, set to temperature) and change it to “Bit 32”.



Keep the Smart Home Manager handy - you will need to reference numbers of the set flags later in Node-RED.

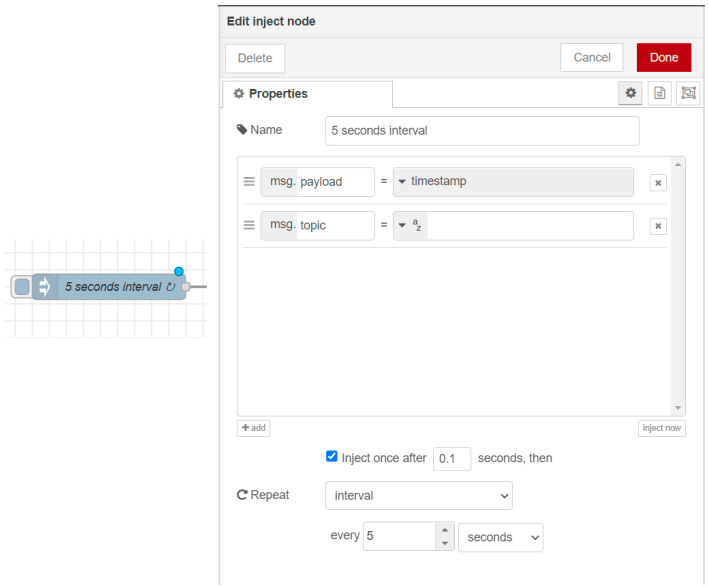
Node-RED configuration

Open Node-RED and log in using your credentials.

Inject node configuration

The Inject nodes allow you to inject messages into a flow, either by clicking the button on the node, or setting a time interval between injects. We will use the latter setting for the purpose of Netio integration.

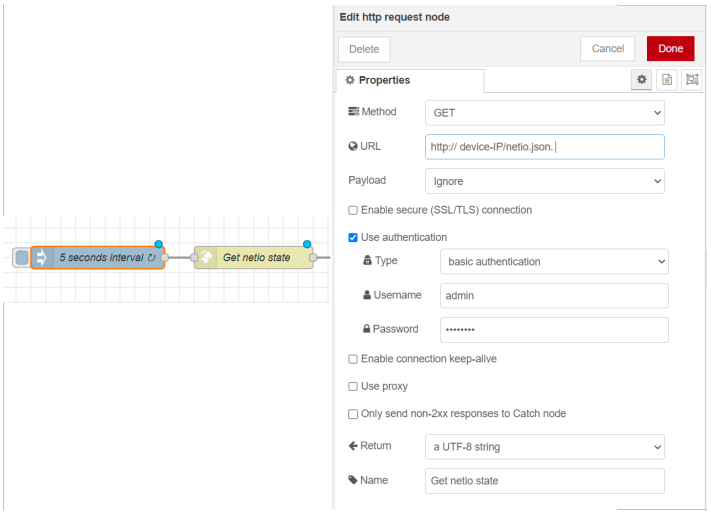
First, drag an inject node from the pallette onto the workspace in Node-RED. Then, edit the node and set up 5-second intervals so that you can get the status of outputs and information about consumption every 5 seconds.



Remember to add relevant names to the nodes you configure so that you can navigate between them easier.

HTTP request node configuration

The next node to add to the workspace is an HTTP request node. Open its properties and set the *Method* to *GET* and paste **http://device-IP/netio.json** into the *URL* field. Then, set the *Payload* to *Ignore*. Use basic authentication as the authentication type. The required Username and Password are your Netio login credentials.



JSON node

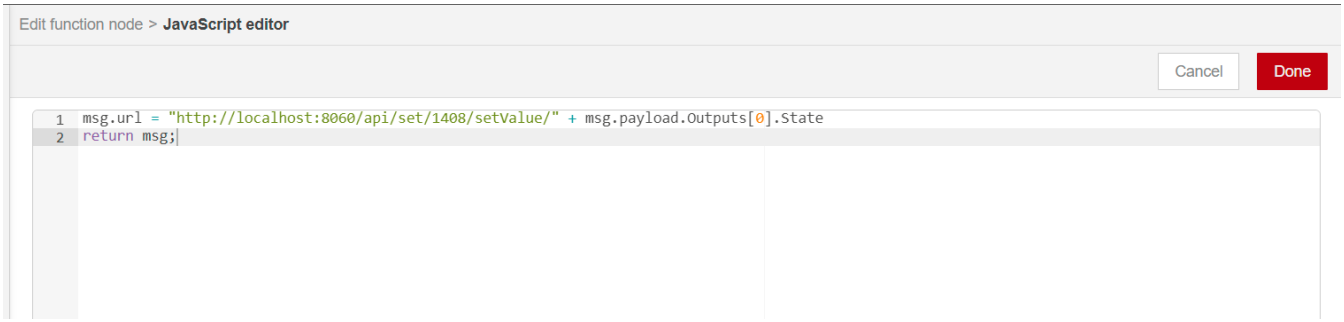
Next, you will need to add a JSON node, which is used to convert between a JSON string and a JavaScript object. The JSON node will, by default, detect what it is being given to convert.

Function node configuration

The Function node allows JavaScript code to be run against the messages that are passed through it. You will use the function nodes to receive information about the Netio outputs' state. Start adding the function nodes onto your workspace and configure them as follows:

In the *On Message* tab, add:

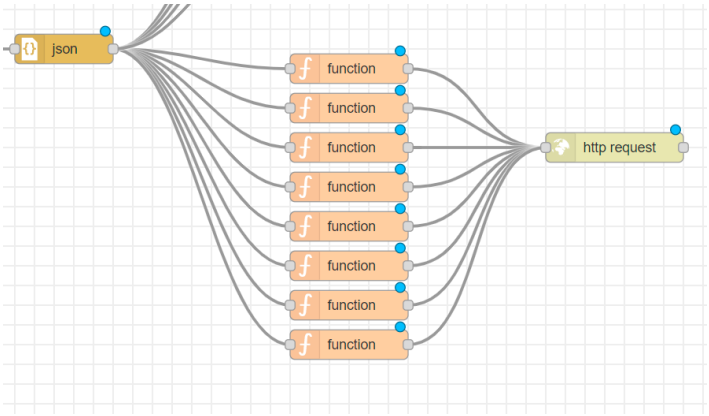
```
msg.url = (
    "http://localhost:8060/api/set/id/setValue/"
    + msg.payload.Outputs[0].State
);
return msg;
```



The above-mentioned will return the sate of Output no. 1 of your NETIO device. In the same fashion, Outputs[1] in the code will provide the sate of NETIO Output no.2, etc.

HTTP request node configuration - units of measure

This node must be added now, if you want to read the measures of such electricity units as voltage, consumption, etc. on your NETIO device integrated with Ampio.

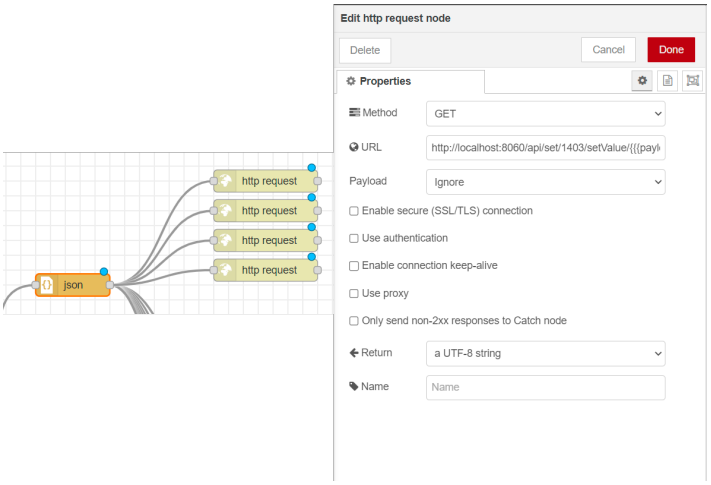


The configuration is performed in a similar fashion described in the previous section, with the following exceptions:

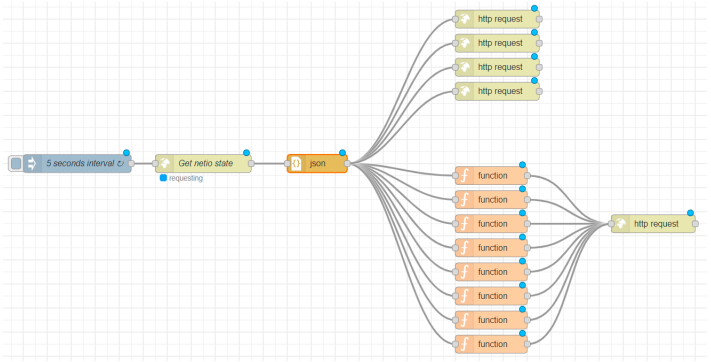
1. You will not use authentication this time

2. Paste the following URL into the URL field:

```
__http://localhost:8060/api/set/1403/setValue/{{{payload.GlobalMeasure.Voltage}}}__
```

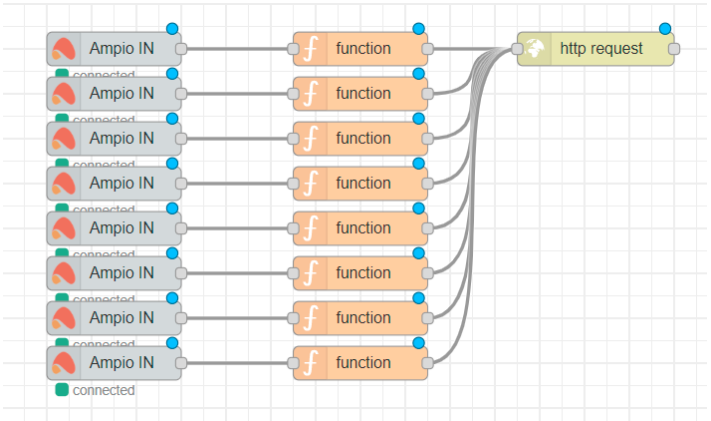


The URL example above will get you the state of the voltage use, but you can replace it with other keywords from your NETIO JSON configuration to obtain readings of other electricity units, like current, load, etc.



Ampio IN node

Finally, for you to be able to control and change the states of NETIO outputs via the Ampio UNI app, you must add the Ampio IN and Function nodes.



In the Ampio IN node, enter the flag number, which you have assigned earlier in Smart Home Manager.

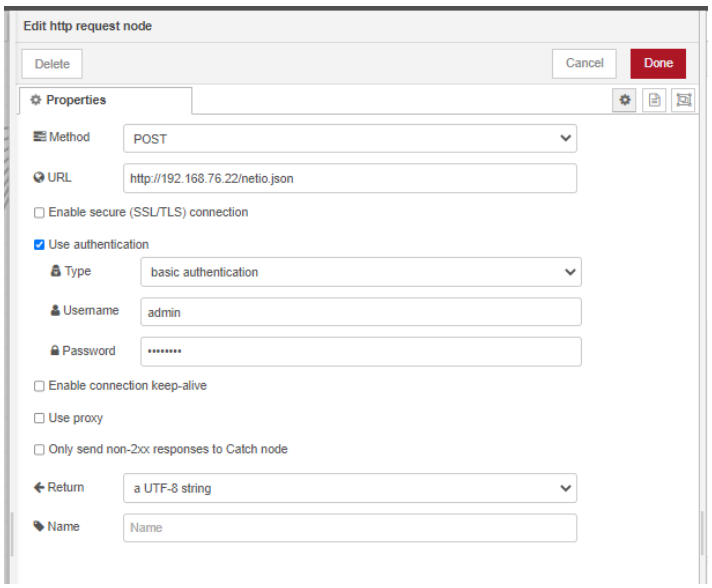
In the Function node enter the following in the *OnMessage* tab:

```
//msg.payload = "{ "Outputs": [{ "ID": 1, "Action": 4 }]}"

msg.payload = { Outputs: [{ ID: 1, Action: msg.payload }] };
return msg;
```

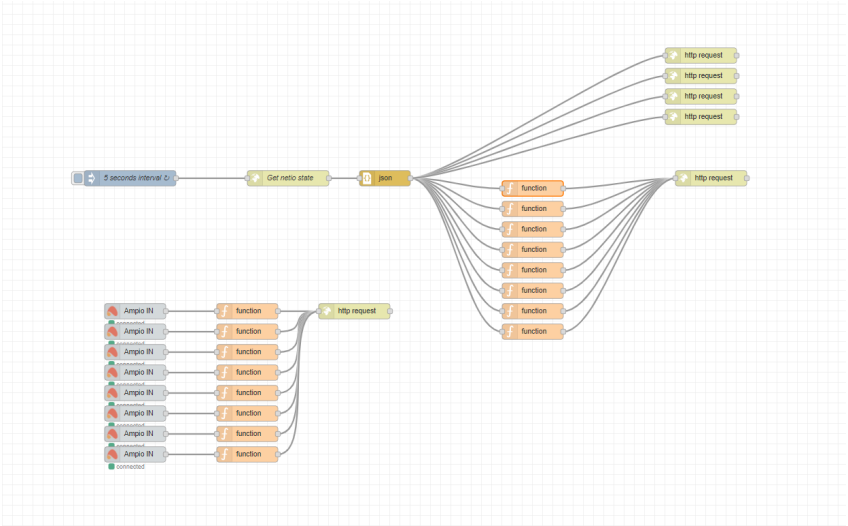
The “ID” value will have to be changed in each function node for each output, for example, Output 1 will be “ID: 1”, Output 2 - “ID: 2, etc.

Configure the http request node in the same manner as you did earlier, but tick also basic authentication where you will enter your NETIO credentials.



Complete Node-RED configuration

The configuration process is now complete. The figure below presents a workspace with concluded configuration for the NETIO PowerPDU to be integrated with the Ampio system.



The last step is to click *Deploy* in Node-RED and the configuration will be saved on the installation’s M-SERV module.

Ampio UNI

In order to finalise this integration process and see its results in the Ampio UNI app, return to the Smart Home Manager and create a new group for “Netio” in the *Grouping* tab that can be found in the left side menu.

Then, select all the objects created in the Adding objects to Smart Home Manager step to the newly created Netio group.

Save and start managing your NETIO device from the Ampio UNI app.