
NETIO Push - JSON

NETIO M2M API protocols docs

Protocol version: Netio Push - JSON Version 6 (JSON ver. 2.0)

30.11.2020

Short summary

Netio Push – JSON is implemented as M2M API protocol, where NETIO device is pushing (sending) status in JSON structure to the server over HTTP(s). The server can respond with JSON structure to control the outputs. It allows output status **monitoring and control** of NETIO power outputs (power sockets 230V or power outlets IEC-320 110/230V).

- This protocol includes power consumption values.
- The Netio Push protocol must be enabled first in the WEB configuration of the respective device. For details, see the “Device WEB configuration” chapter.

The status file is pushed (send) over HTTP(s)

- 1) Periodically based on defined Push Period
- 2) Immediately when the measured value exceeds defined delta value since the last push

Supported devices

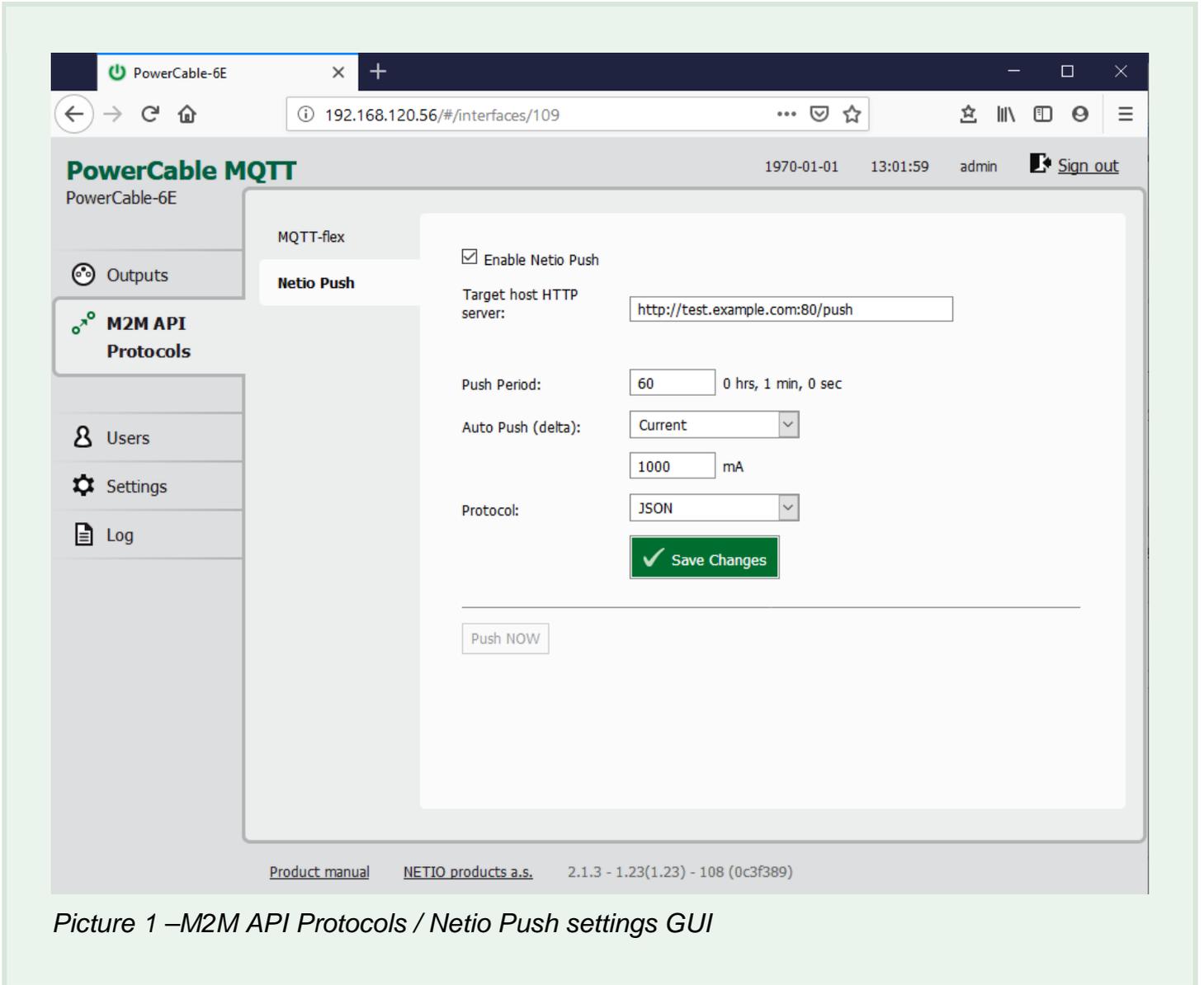
- NETIO PowerCable REST 101x
- NETIO PowerPDU 4PS
- NETIO PowerDIN 4PZ
- NETIO PowerBOX 3Px

Supported firmware

3.0.0 and later

Device WEB configuration

M2M API protocol can be configured over the web administration – select “M2M API Protocols” in the left-hand side menu and then select the “Netio Push” tab.



Picture 1 –M2M API Protocols / Netio Push settings GUI

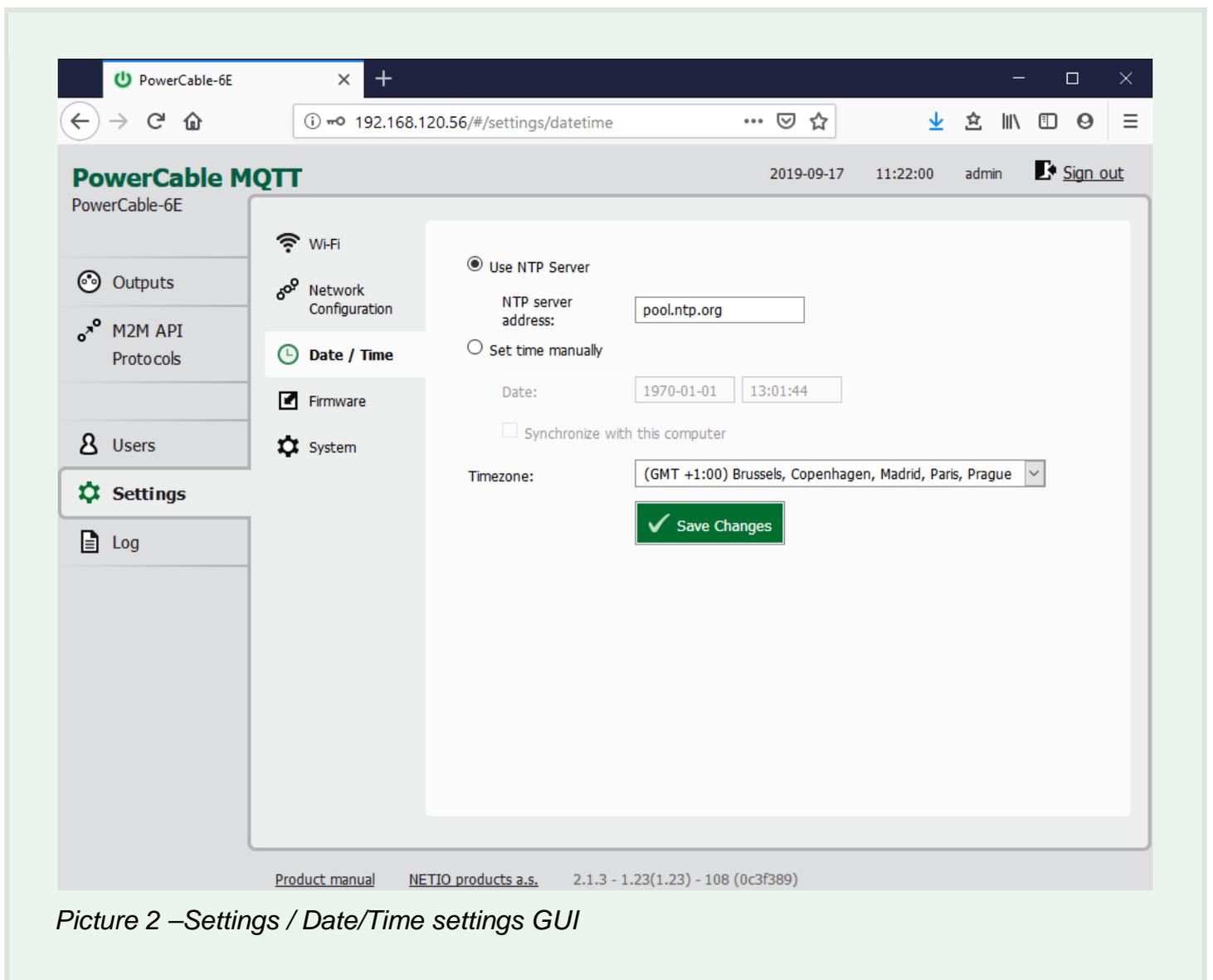
- **Enable Netio Push** – Enable/disable the M2M API protocol
- **Target host HTTP server** – definition of target parameters
 - http / https to define protocol
 - Target URL address
 - Port
 - Username
 - Password

Example:

`https://username:password@example.com:8445/netio`

- **Push Period** – period for automatic send (Push) of data in seconds
- **Auto Push (delta)** – settings for immediate data send (Push) when the measured value exceeds a limit
 - Parameter – Current only for now
 - Value – a limit of the selected parameter
- **Protocol** – selection of protocol used for Push
 - JSON
 - XML (not described in this document)

The status file contains “Date and time” of the NETIO device. We recommend set up NTP date and time synchronization. It can be done over the web administration – select “Settings” in the left-hand side menu and then select the “Date / Time” tab. Otherwise the time is reset after device power off.



Picture 2 –Settings / Date/Time settings GUI

-
- **Use NTP server** – Use to synchronize Date and Time with NTP server
 - **Set time manually** – Use to set up Date and Time manually. The Date and Time will be set to 1970-01-01 12:00:00 (and Timezone shift) after every power outage
 - **Date** – Date and Time used in PowerCable. It can be edited if “Set time manually” selected.
 - **Synchronize with this computer** – Select to take over Date and Time from your Computer.
 - **Timezone** – Select the relevant Timezone

Communication protocol

- **HTTP / HTTPS** selected in the URL
- **Basic authorization** used when Username / Password defined in “Target host HTTP server”
-> Username and password are defined in URL to simplify a setup allowing copy/paste of only one “value”. However, the system in PowerCable extract the **credentials** from the URL and **sends** these **only in the header**.

NETIO JSON protocol structure

JSON standard: RFC4627

status json data:

```
{
  "Agent": { "Model": "4PZ", "DeviceName": "PowerDIN-
EE", "MAC": "24:A4:2C:38:E9:EE", "SerialNumber": "24A42C38E9EE", "JSONVer": "2.3
", "Time": "2020-10-
19T16:58:40+01:00", "Uptime": 11064, "Version": "2.5.5", "OemID": 400, "VendorID"
: 0, "NumOutputs": 4, "NumInputs": 2 },
  "GlobalMeasure": { "Voltage": 238, "TotalCurrent": 82, "TotalLoad": 12, "TotalEner
gy": 36, "OverallPowerFactor": 0.59, "Frequency": 50.05, "Phase": -
37.28, "EnergyStart": "2020-06-30T17:04:48+01:00" },
  "Outputs": [
    { "ID": 1, "Name": "Power output
1", "State": 1, "Action": 6, "Delay": 2020, "Current": 82, "PowerFactor": 0.59, "Phas
e": -37.28, "Energy": 8, "ReverseEnergy": 0, "Load": 12 },
    { "ID": 2, "Name": "Power output
2", "State": 0, "Action": 6, "Delay": 2020, "Current": 0, "PowerFactor": 1.00, "Phase
": 0.00, "Energy": 27, "ReverseEnergy": 0, "Load": 0 },
    { "ID": 3, "Name": "Free Contact 3", "State": 0, "Action": 6, "Delay": 2020 },
    { "ID": 4, "Name": "Free Contact 4", "State": 1, "Action": 6, "Delay": 2020 }
  ],
  "Inputs": [
    { "ID": 1, "Name": "Input 1", "State": 1, "S0Counter": 28 },
    { "ID": 2, "Name": "Input 2", "State": 0, "S0Counter": 30 }
  ]
}
```

Note: A DebugLog data can follow after „Outputs“ section if this is enabled in the Settings/System/Debug Log. Used only for troubleshooting together with PowerCable manufacturer.

Values description

Global values:

"Model": "101x"	<i>Model identification</i>
"DeviceName": "PowerCable 6C"	<i>Device name (user defined on web)</i>
"MAC": "24:A4:2C:38:D4:6C"	<i>The main MAC used as a Serial Number</i>
"JSONVer": "2.0"	<i>Protocol version</i>
"Time": "1970-01-01T13:14:19+01:00"	<i>Date and time of the NETIO device</i>
"Uptime": 859	<i>[s] The Uptime value</i>
"Version": "2.1.3"	<i>Firmware version</i>
"OemID": 7	<i>Manufacturer internal use</i>
"VendorID": 0	<i>Manufacturer internal use</i>
"NumOutputs": 1	<i>Number of outputs</i>
"Voltage": 235.8	<i>[V] Instantaneous voltage</i>
"TotalLoad": 1	<i>[W] Total Power of all power outputs</i>
"TotalEnergy": 965	<i>[Wh] Instantaneous value of the Total Energy counter</i>
"OverallPowerFactor": 0.22	<i>[-] Instantaneous True Power Factor weighted average from all meters</i>
"Frequency": 49.9	<i>[Hz] Instantaneous frequency</i>
"EnergyStart": "2017-06-23T16:47:53+01:00"	<i>Date and time of the last reset of all energy counters</i>

Values for specific output (example values below are for output 1):

"ID": 1	<i>Output number</i>
"Name": "Power output 1"	<i>Output name (user defined on web)</i>
"State": 1	<i>Output state</i>
"Action": 6	<i>Output action (6 = Ignored value, only placeholder)</i>
"Delay": 2000	<i>[ms] Output delay for short On/Off</i>
"Current": 0	<i>[mA] Instantaneous current of the output</i>
"PowerFactor": 1.00	<i>[-] Instantaneous True Power Factor</i>
"Energy": 114099	<i>[Wh] Instantaneous value of the Energy counter</i>
"Load": 0	<i>[W] Total Power of the output</i>

Values for specific input (example values below are for input 1):

"ID": 1	<i>Input number</i>
"Name": " Input 1"	<i>Input name (user defined on web)</i>
"State": 1	<i>Input state (0= open/off, 1= closed/on)</i>
"S0Counter": 28	<i>Input S0 counter value</i>

Output control

Device is periodically connecting to the Target host http(s) server within push period (Picture 3 – default 60seconds). It means the output can be switched only when the TCP/IP connection is established. In default once every 60 seconds.

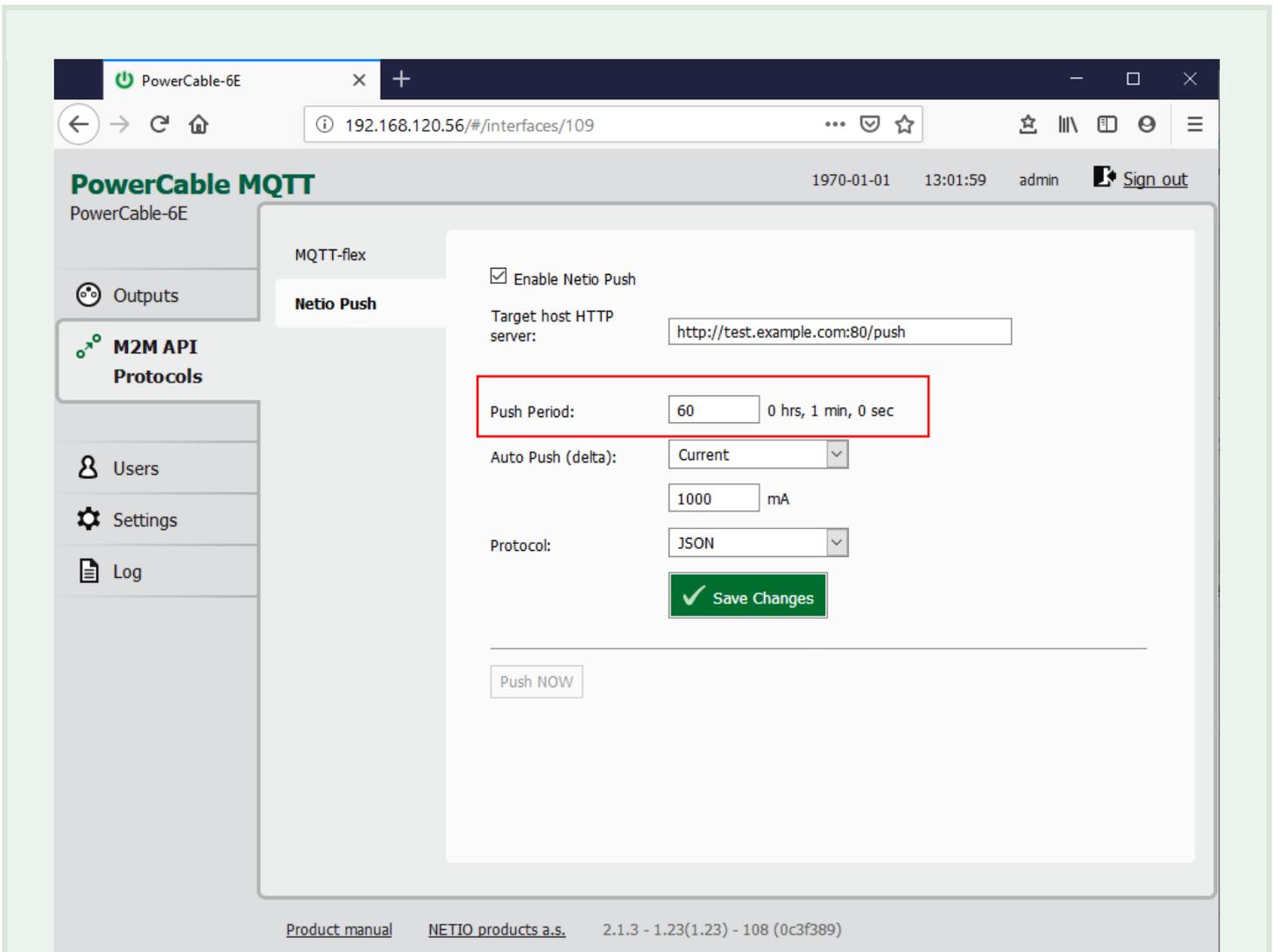
If the response from Target host is error code 200 (OK) and the payload contains valid NETIO JSON structure with control command then the Output will be set accordingly.

This way is possible to control device even if this is in private network behind NAT.

The NETIO JSON structure is described in section above and tag “Action” or “State” to be used for controlling the respective output.

Output Actions

- **0** – Turn **OFF**
- **1** – Turn **ON**
- **2** – Short OFF (restart)
- **3** – Short ON
- **4** – Toggle (invert the state)
- **5** – No change
- **6** – Ignored (return value from reading the tag)
Actual output value is in “State” tag (0 / 1).



Picture 3 – M2M API Protocols / Netio Push settings GUI – Push Period

Examples of JSON returned by Target host:

Switch Power output 1 to ON by Action tag:

```
{
  "Outputs": [
    {
      "ID": 1,
      "Action": 1
    }
  ]
}
```

or (State tag value will not be reflected)

```
{
  "Outputs": [
    {
      "ID": 1,
      "State": 0,
      "Action": 1
    }
  ]
}
```

or by State tag (Action tag must have value 6)

```
{
  "Outputs": [
    {
      "ID": 1,
      "State": 1,
      "Action": 6
    }
  ]
}
```

Switch Power output 2 to ON for 15 seconds, then switch it OFF.

```
{
  "Outputs": [
    {
      "ID": 2,
      "Action": 3,
      "Delay": 15000
    }
  ]
}
```

Command to control more outlets:

Switch Power output 1 to ON, Toggle Output 2 and Switch Output 4 to ON for 15 seconds:

```
{
  "Outputs": [
    {
      "ID": 1,
      "Action": 1
    },
    {
      "ID": 2,
      "Action": 4
    },
    {
      "ID": 4,
      "Action": 3,
      "Delay": 15000
    }
  ]
}
```

Note: The JSON can include also "Agent" and "GlobalMeasure" sections but these would be ignored.

Document history

Document Revision	Publication Date	Description
1.0	16.9.2019	Initial release – Netio Push JSON, Version 5 (JSON ver. 2.0)
1.1	19.11.2020	Added Output Control (Version 6) and other supported devices