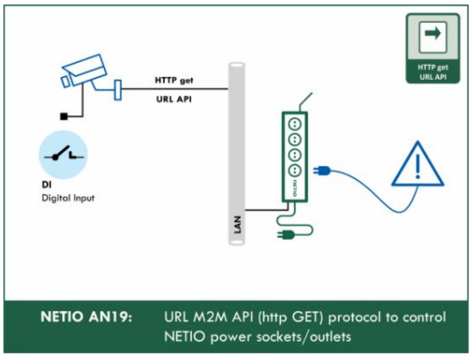
Server Room®
environments

How to use an API to Remotely Control Smart Power Sockets


HTTP get
URL API
HTTP get
URL API
DI
Digital Input
LAN
NETIO
NETIO AN19:
URL M2M API (http GET) protocol to control NETIO power sockets/outlets

This application note describes how to use a ‘URL API’ or ‘http get’ machine 2 machine (M2M) API protocol to control the power outlets on a compatible NETIO device. With a specific URL, an electrical socket can be switched ON/OFF/TOGGLE or Short PULSE. Compatible devices include the POWERPDU-4C, PowerCable Rest and PowerBox 3Px models.

An Application Programming Interface (API) is a computer interface built into components and systems for external communication and control. The API defines the allowed data calls or requests, how to make them, data formats and conventions to follow.

NETIO compatible devices with outlet sockets can be controlled with URL M2M API calls (accessing a URL with parameters) in two ways:

- Using a custom Lua script (additional document)
- Using the standard URL API (http GET)” M2M API (this document)

The difference in the methods Lua script and M2M API, is that the URL string and following reactions in a standard M2M API cannot be modified by the user.

The string of characters received by NETIO compatible devices in the URL is split into individual commands and the device then sets its outputs to the desired states. The Action numbers correspond to the numbering in all M2M protocols.

Output actions:

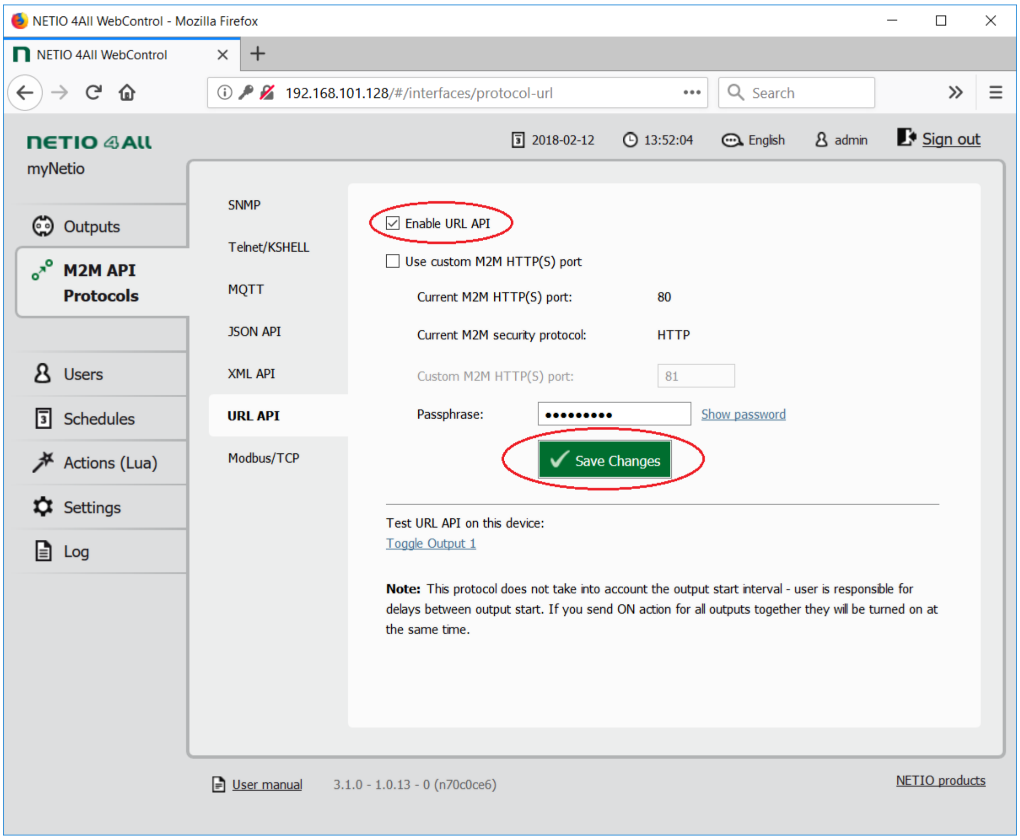
- 0 = Output switched off (OFF)
- 1 = Output switched on (ON)
- 2 = Output switched off for a short time (short OFF)
- 3 = Output switched on for a short time (short ON)
- 4 = Output switched from one state to the other (TOGGLE)
- 5 = Output state unchanged (no change)

URL API format example (toggle output 1):

Available upon request

1. Configuring NETIO 4x smart power sockets

In the NETIO 4x web administration, go to the M2M API Protocols – URL M2M API section, enable URL API and set a passphrase. Modify the port number if needed. Click Save Changes. After saving the changes, the device restarts (about 1 minute).



☑ Enable URL API

□ Use custom M2M HTTP(S) port

Current M2M HTTP(S) port: 80

Current M2M security protocol: HTTP

Custom M2M HTTP(S) port: 81

Passphrase: ••••••••• Show password

Test URL API on this device:
Toggle Output 1

Note: This protocol does not take into account the output start interval - user is responsible for delays between output start. If you send ON action for all outputs together they will be turned on at the same time.

2. Creating the command string

To control the sockets, create a command string in the form of: http://<netioIP>/netio.cgi?pass=<Passphrase>&output<X>=<action>

Replace <netioIP> with the IP address of your NETIO smart sockets. If you use a port different from the default (80), append a colon and the port number to the NETIO IP.

Example (port 8080): 192.168.101.118:8080

2.1 Parameters

2.1.1 pass=<Passphrase>

Passphrase specified in M2M API Protocols – URL M2M API

The default Passphrase is the MAC address that is shown on the package label. For security reasons, we recommend changing it.

Example (the Passphrase is set to netio-psw): **pass=netio-psw**

If a passphrase is not required, this parameter can be left empty. It can be omitted completely from the command string, or it can be present but empty (only pass=).

2.1.2 output<X>=<action>

- X
  - Specifies the output number (1 to 4)
  - Example – to specify output 2: output2

2.2.3 action

- Specifies the action to perform with the output.
- 0 – turns the output OFF
- 1 – turns the output ON
- 2 – turns the output OFF for a short time (if the output was off, it will be turned on)
- 3 – turns the output ON for a short time (if the output was on, it will be turned off)
- 4 – toggles the current output state
- 5 – leaves the output unchanged

Example – to turn on output 3: **output3=1**

To control more outputs at the same time, separate individual parameters with the **&** sign.

Example – to turn on output 2, turn off output 3 and toggle output 4:
output2=1&output3=0&output4=4

2.1.4 delay<X>=<time(ms)>

Specifies the pulse duration for actions 2 and 3.

If it is not specified or the value is 0, the value specified in the web administration is used.

Replace **<X>** with the output number and **<time(ms)>** with the time in milliseconds (the minimum value is 100).

Example – to specify the pulse duration for output 3 to 500ms: **delay3=500**

Example of a complete command string: **http://address?pass=netio-psw&output3=2&delay3=500**

3. Return values

The following responses to the http get request are possible:

200 OK

Everything went OK

400 Bad Request

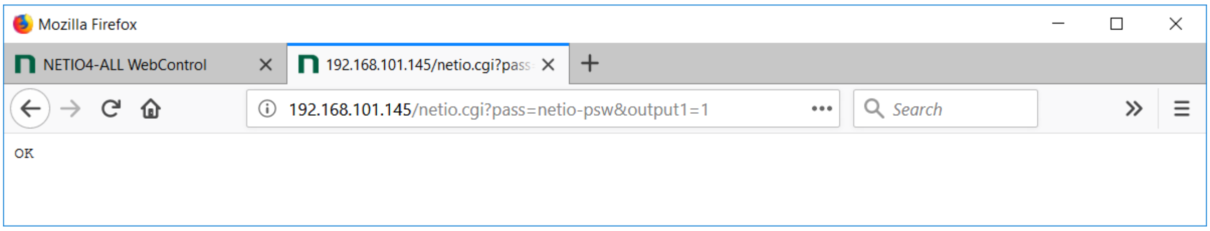
Incorrect format of the command string

401 Unauthorized

Incorrect Passphrase

403 Forbidden

URL M2M API is disabled on the NETIO device (see Configuring NETIO 4x)

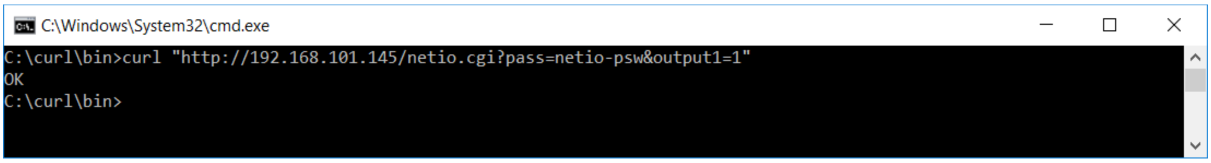


```
192.168.101.145/netio.cgi?pass=netio-psw&output1=1
```

```
OK
```

4. Controlling the socket from the command line

For command line control, a utility such as cURL (available for download from http://curl.haxx.se/download.html) or any other command that can access a URL (such as a web browser) can be used.



```
C:\curl\bin>curl "http://192.168.101.145/netio.cgi?pass=netio-psw&output1=1"
OK
C:\curl\bin>
```

4.1 Windows

Download the cURL utility, open the command line (CMD) and go to the bin folder in the curl installation folder.

The command string needs to be enclosed in double quotes and preceded with curl.

Example: turn on socket 1: curl *"http url"*

4.2 Mac OS X a Linux

By default, cURL is installed in your operating system. The command to control the outputs is the same as in Windows, see above.

5.0 Frequently Asked Questions (FAQs)

1. Is it possible to control several outputs at once with a single command?

Yes, one command string can contain multiple commands to control the outputs. These need to be separated with the & sign.

2. After entering the command string, I am redirected to the web administration. What should I do?

Probably a different port than 80 was set. For ports different than 80, the IP address must be followed by a colon and the port number. If the problems persist, please contact customer support.

3. I turn on the socket using the URL but then the Scheduler is supposed to control the same socket. Will the time sequence work as expected?

Yes, after executing a URL M2M API command, the socket can still be controlled via all other channels (Scheduler, WatchDog, mobile app, button, ...)

4. Which method does the URL-based CGI M2M API use, http post or http Get?

This is the http Get method.

5. What is the difference between AN04 and AN19?

Both Application Notes describe similar ways of controlling NETIO smart sockets with a URL. The difference is in the command syntax and its processing.

- The AN19 Application Note uses a standard M2M API. It is easier for the user, but the behaviour cannot be customized.
- The AN04 example uses a custom Lua script. In this case, the command syntax and NETIO behaviour is determined by the custom script. The AN04 itself is an example of use rather than a protocol definition.

6. Is it possible to read the output states using URL M2M API Get?

No. We recommend Telnet or XML-based protocols of the M2M API.

6. As a response to the URL command, all I get is a HTML page with a simple "OK". Is it possible to customize this response?

Not at this moment. In the future, we plan to add a possibility to specify a return code in case of the standard M2M API (AN19) or a custom string in case of the Lua script (AN04).

7. Supported FW versions:

3.0.1 and higher

This Application Note is compatible with:

7.1 PowerPDU 4C a small PDU with power measurement

PowerPDU 4C is a small 110/230V PDU (Power Distribution Unit). Each of the four IEC-320 C13 outlets can be independently controlled (ON / OFF / RESET / TOGGLE). Electrical parameters (A, W, kWh, TPF, V, Hz) are measured with high accuracy at each outlet. The device features two LAN ports (and a built-in Ethernet switch) for connecting to a LAN. Each power output supports ZCS (Zero Current Switching) to protect the connected equipment.

https://www.serverroomenvironments.co.uk/netio-powerpdu-4c-metered-pdu

7.2 PowerCable REST 101x

PowerCable REST 101x is a smart Wi-Fi power socket for integration with third-party systems using an open API. PowerCable REST provides electrical measurements and control of the output sockets using one of three http-based REST protocols - XML, JSON or URL API.

There are two features models:

- IEC320 sockets
  https://www.serverroomenvironments.co.uk/netio-smart-wifi-powercable-iec320-socket-rest-api
- BS1363 UK style sockets
  https://www.serverroomenvironments.co.uk/netio-smart-wifi-powercable-bs1363-uk-socket-rest-api

7.3 PowerBox 3Px

PowerBox 3PG is a smart power strip with 3 BS-Style square pin output sockets and an RJ45 Ethernet LAN connection port. Each output socket can be switched ON/OFF using a web interface. The PowerBox 3PG has an open API and can integrate with third-party systems using standard Machine2Machine (M2M) protocols including JSON, MODBUS/TCP, SNMP, MQTT-flex, XML and Telnet.

https://www.serverroomenvironments.co.uk/powerbox-3pg-smartpower-strips

8. Document Number

SRE-AN003-edition-1