
URL API / HTTP(s) GET

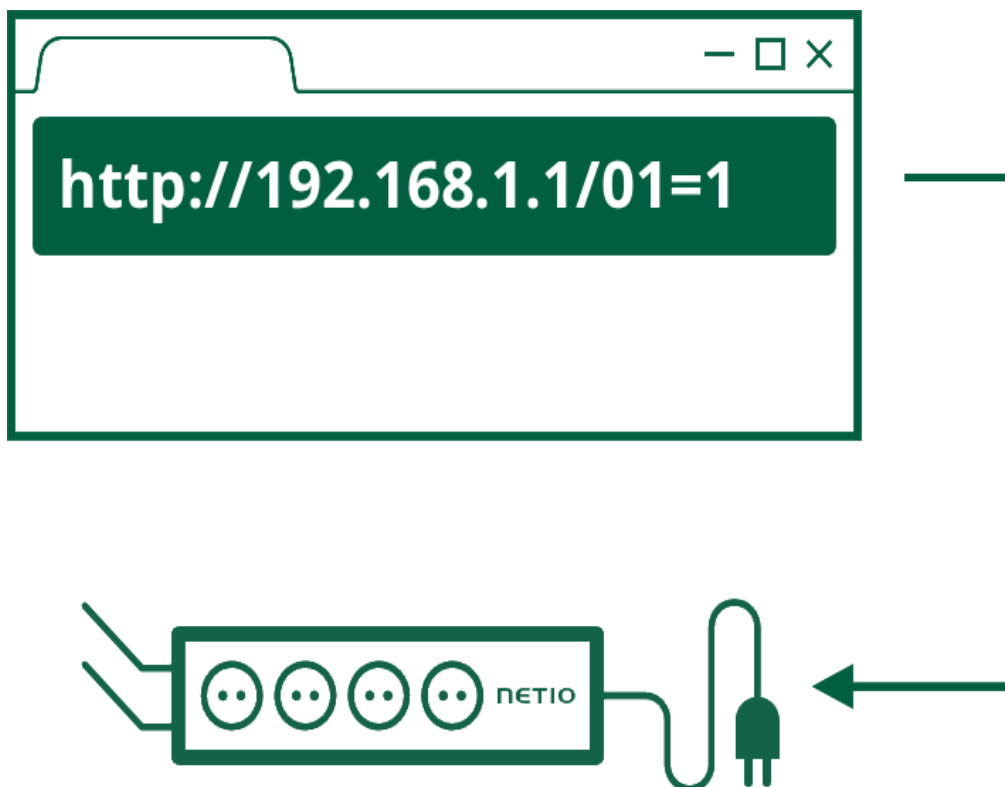
NETIO M2M API protocols docs

7.4.2021

Short summary

URL API protocol is one way – control only, command based M2M API protocol, where NETIO device is HTTP(s) server accepting commands sent over HTTP(s) GET to control NETIO power outputs. URL API offers very easy protocol to control NETIO outputs by calling defined address.

- URL API protocol must be enabled first in the WEB configuration of the device. For details, see the “NETIO WEB configuration” chapter.
- This protocol is HTTP or HTTPS based. Not all NETIO devices supports HTTPS.



Supported devices

NETIO 4x Linux based devices: networked power sockets with LAN / WiFi connectivity.

- NETIO PowerPDU 4C
- NETIO 4, 4All (obsolete products)

NETIO 4x firmware – 3.0.1 and later

Standard NETIO devices: networked power sockets with LAN / WiFi connectivity.

- NETIO PowerCable REST 101x
- NETIO PowerBOX 3Px
- NETIO PowerBOX 4Kx
- NETIO PowerDIN 4KZ
- NETIO PowerPDU 4PS
- NETIO PowerPDU 8QS

Note: *Firmware – 3.0.1 and later,
There have to be enabled Telnet protocol (1 protocol only supported in one time)*

URL API integration in NETIO PowerPDU 4C product

URL API as protocol in NETIO can be processed by device in a two different ways:

- 1) **M2M API Protocol** – fixed commands, described in this document.
You can find application examples in the “**NETIO AN19** URL M2M API (http GET) protocol to control NETIO power sockets/outlets”
- 2) **URL request processed in Lua script** – user definable commands interacting with Lua scripts. Parameters can be modified and user script can be programmed. URL request managed in Lua use the same http port as web administration.
*URL in Lua is not described in this document - check the “**NETIO AN04** Remote power socket control by URL (http get handled in Lua script)” for more details.*

*This document provides basic info about the M2M API protocol.
Other device functions are described in the product manual*

Quick start with URL API & NETIO

There is no read function available in URL API M2M Protocol.

URL API Control commands (HTTP GET)

```
http://<netioIP>/netio.cgi?pass=<Passphrase>&output<X>=<action>
```

X is number of output

outputX= value of supported action: 0 - off, 1 - on, 2 - short off, 3 - short on, 4 - toggle, 5 - no change

Example: Switch Power output 1 to ON:

```
http://192.168.120.136/netio.cgi?pass=01234567890AB&output1=1
```

PowerPDU 4C: HTTP(s) port

For PowerPDU 4C device there are 2 different HTTP(s) ports:

- 1) The web administration of the device - HTTP(s).
Web administration is in the Settings/System (HTTP) or System/Security Settings (HTTPs).
- 2) Separated HTTP(s) port for the M2M API protocols (XML, JSON, URL API).
Web administration is in each M2M API protocol settings.

All HTTP(s) protocols (XML / JSON / URL) share one HTTP(s) port.

Web Configuration	M2M API	
HTTP	HTTP	Possible
HTTPs	HTTPs	Possible
HTTP	HTTPs	Not possible today
HTTPs	HTTP	Not possible today

General NETIO 4x output functions

Output actions – “write” function

- **0** – Turn **OFF**
- **1** – Turn **ON**
- **2** – Short OFF delay (restart)
- **3** – Short ON delay
- **4** – Toggle (invert the state)
- **5** – No change

Short ON / OFF delay

This command switches a power output On / Off for a defined time. It is useful for example to power-cycle a server with a defined switch-off time, or to switch on a pump for a defined time.

This “short” delay is protected: the power output will remain in the defined state regardless of any other M2M requests received. During this time, the output state can only be changed by pressing the button on the NETIO device and this action cancel M2M short ON/OFF command for the particular output. Other requests to control the particular output are simply ignored and an ERROR logged with reason rejected in a device Log.

The short ON / OFF delay interval can be defined in the device web administration. It is specified in ms (milliseconds) and rounded up to hundreds of milliseconds (0,1s).

This interval can be also defined using some M2M API protocol commands. In that case, it is valid only for a single protocol session (the following short ON / Short OFF command). When the connection is closed or restarted, the interval is reset to the device default value (defined in the web administration for each output).

Security issues

Do not use default usernames and passwords! Keep your Ethernet and WiFi networks secured.

Power-Up outputs state

All outputs are Off during the first 25 to 30 seconds after power-up.

After this time, all outputs are set to the selected state:

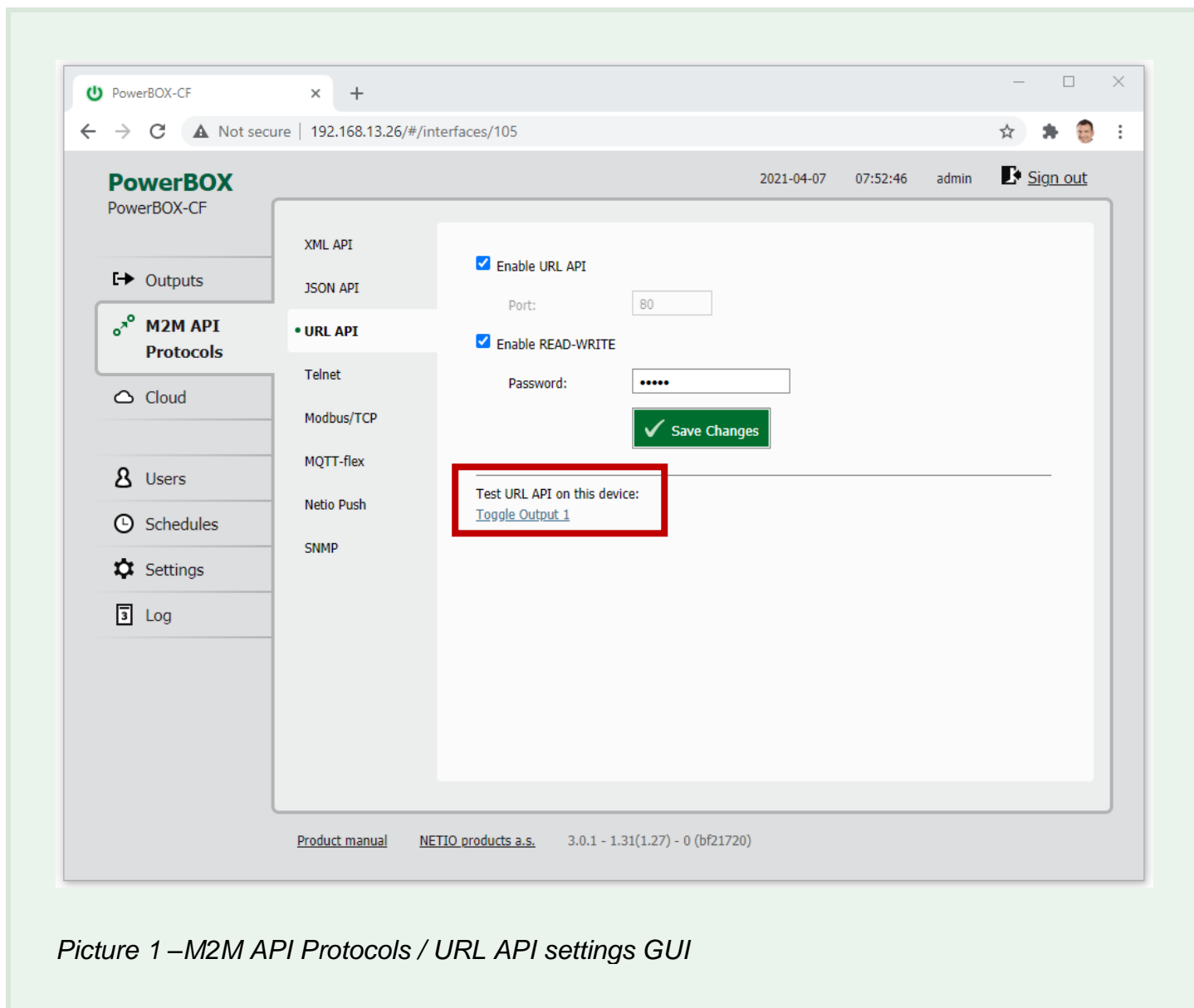
- **Last Output state**
After a power outage, the NETIO device sets each power output to the last stored state of this one output. The current state of each power output (socket/power outlet) is internally stored every 8 seconds.

Note: **Function Scheduler** is checked in Power-Up initialization. When enabled, it can affect one or more power output stated based on current time and date.

Custom based **Lua scripts** can affect output stated too.

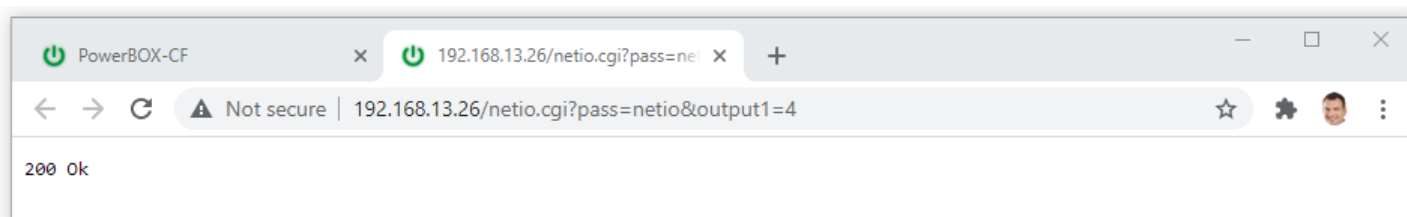
NETIO WEB configuration

M2M API protocols can be enabled and configured only over the web administration – select “M2M API Protocols” in the left-hand side menu and then select the “URL API” tab.



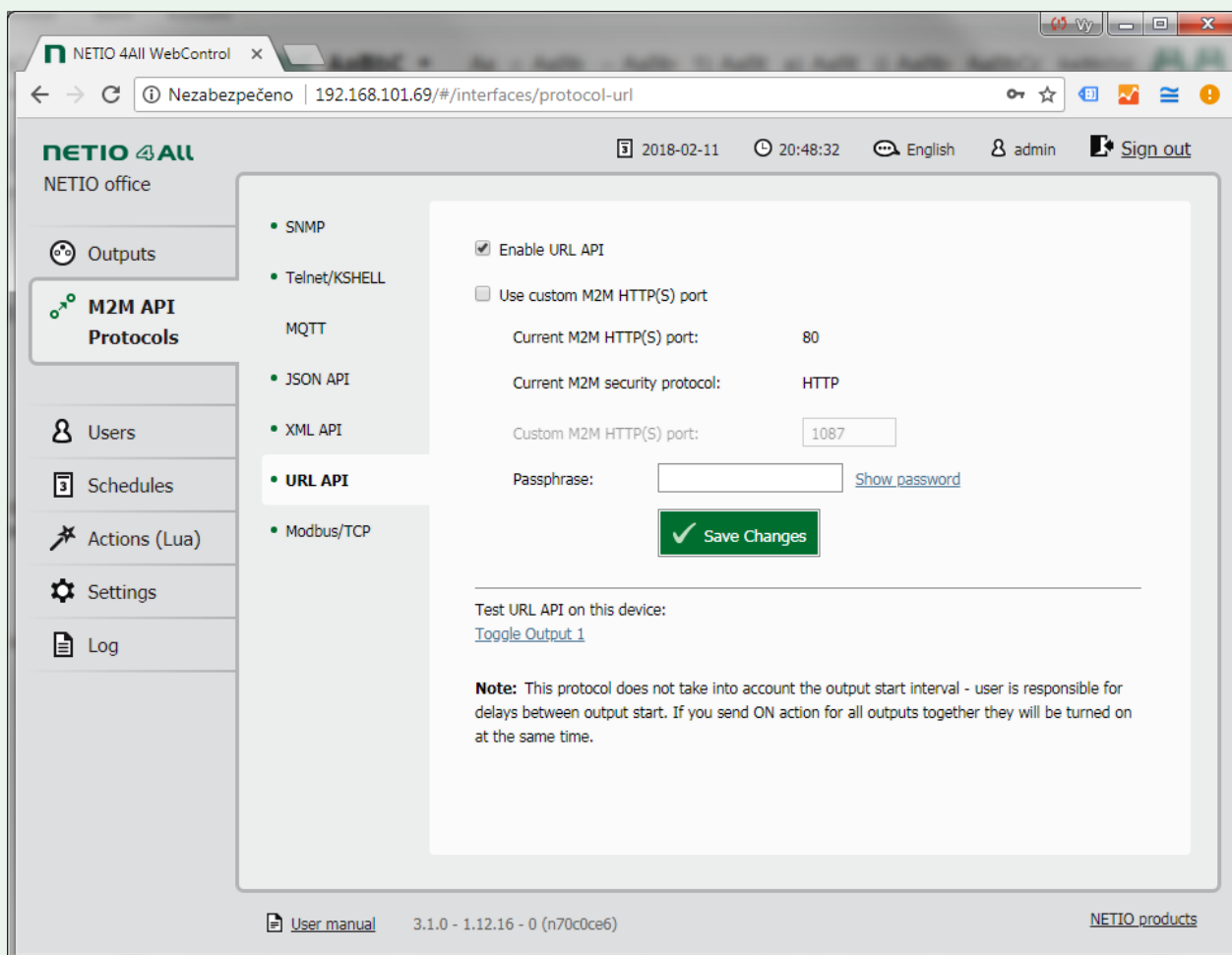
Picture 1 –M2M API Protocols / URL API settings GUI

- **Enable URL API** – Enable/disable the M2M API protocol
- **Port** – HTTP port defined for the device
- **Enable READ-WRITE** (read is not possible for URL API)
- **Password** – Password for commands (default: “netio”, small letters) – required in URL
- **Check settings** by Toggle Output 1 link:



PowerPDU 4C WEB configuration

M2M API protocols can be enabled and configured only over the web administration – select “M2M API Protocols” in the left-hand side menu and then select the “URL API” tab.



Picture 2 –M2M API Protocols / URL API settings GUI for PowerPDU 4C

- **Enable URL API** – Enable/disable the M2M API protocol
- **Use custom port** – Check to enable custom port setting
 - **Current port** –Currently used port
 - **Custom port** – Custom port set for XML / JSON / URL API protocols
- **Passphrase** – Password for commands (default: MAC address without colons, small letters)

Notes

- **The device webserver is restarted after Saving of URL API settings.**
- Empty Passphrase means no authorization and any passphrase is accepted in URL comm

NETIO URL API protocol structure

There is **no read function** available in the NETIO URL API M2M Protocol.

URL Control command – HTTP(s) GET request

```
http://<netioIP>/netio.cgi?pass=<Passphrase>&output<X>=<action>
```

X is number of output

- **outputX=** value of supported action:
0 - off, 1 - on, 2 - short off, 3 - short on, 4 - toggle, 5 - no change
- **delayX=** value (ms) of delay for a short off/on action. If a delay parameter is not listed along with a short off/on action command then a delay defined in web administration is used.

Switch Power output 1 to ON:

```
http://<netioIP>/netio.cgi?pass=<Passphrase>&output1=1
```

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

```
http://<netioIP>/netio.cgi?pass=<Passphrase>&output2=2&delay2=15000
```

More outputs control command:

Switch Power output 1 to OFF, Invert state of Output 2 and Switch Output 3 to ON for 900 ms:

```
http://<netioIP>/netio.cgi?pass=<Passphrase>&output1=0&output2=4&output3=3&delay3=900
```

Status codes

Status code	Response content (plain text)	Description
200 OK	OK	Authorized and command executed
302 Moved Temporarily	<i>See note 1 below table</i>	HTTP request sent while HTTPS is enabled on NETIO device
400 Bad Request	Bad Request	Control command syntax error
401 Unauthorized	Unauthorized	Invalid Passphrase
403 Forbidden	Forbidden	URL API is not enabled

Note 1 – status 302 response content:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
```

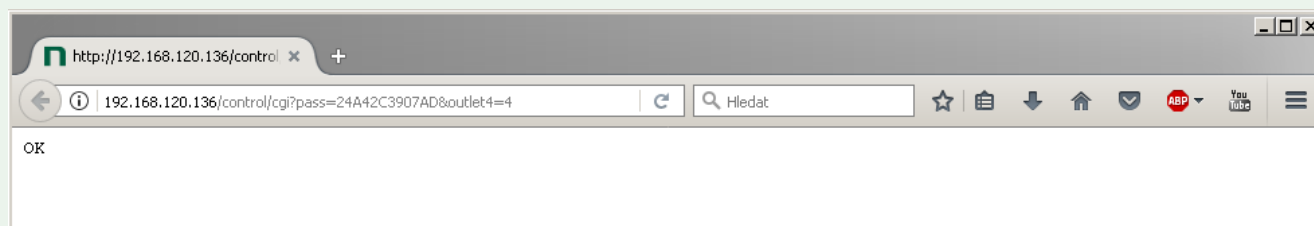
```
"http://www.w3.org/TR/html4/strict.dtd"><html><head><title>Document not found</title></head><body><p>Web interface moved to HTTPS.</p></body></html>
```

Request in Web Browser -> redirection to https web administration

Response content – interpretation by Web Browser:

```
<html>
  <head></head>
  <body>
    <pre>OK</pre>
  </body>
</html>
```

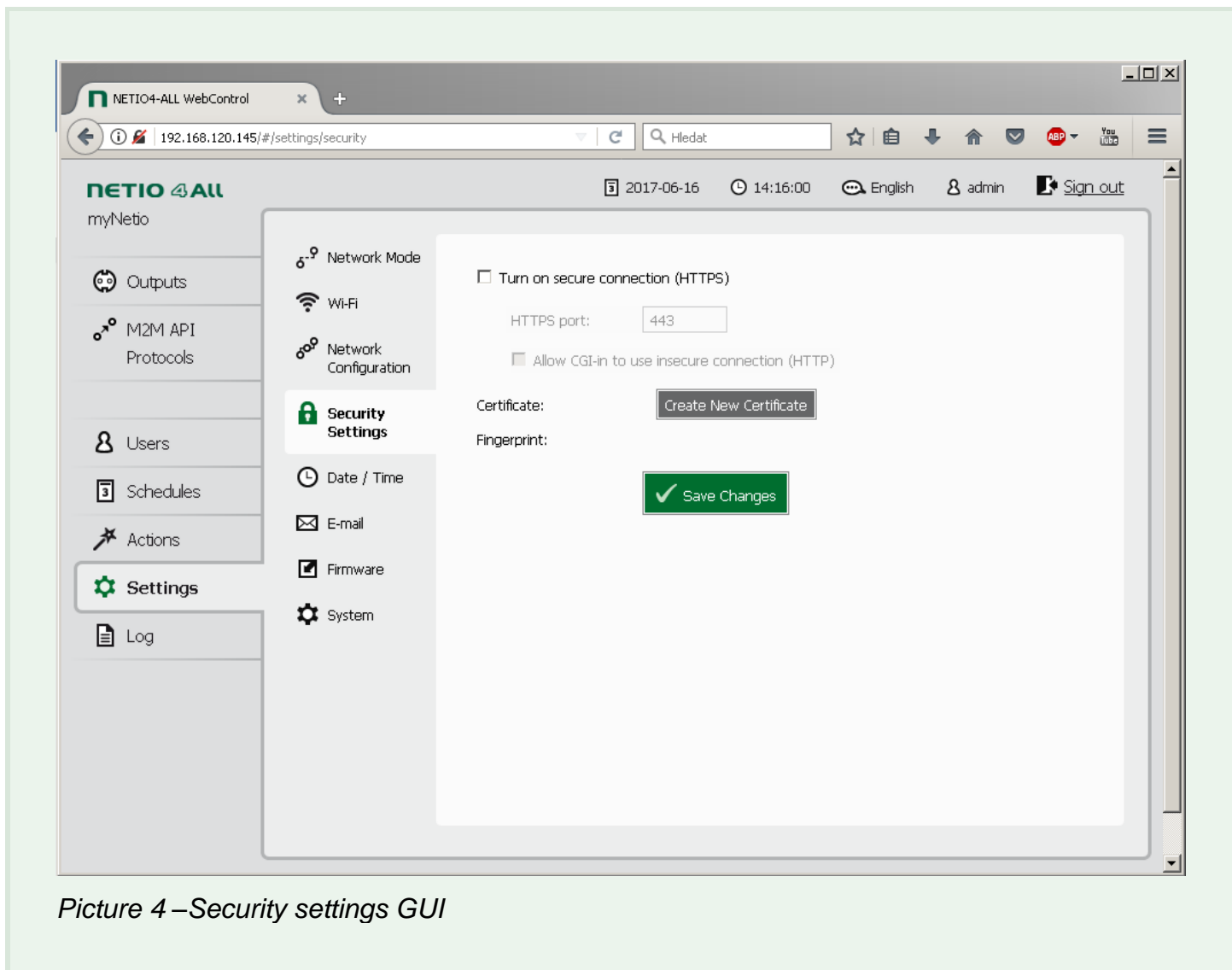
Response content example:



Picture 3 – Response content example “OK”

HTTPs = secure connection for PowerPDU 4C

NETIO PowerPDU 4C can use secure connection (HTTPs) for web administration and HTTP-based M2M API Protocols (JSON, XML, URL). This security feature can be enabled in the web administration: Settings / Security Settings. After checking “Turn on secure connection (HTTPs)” and saving the changes, only secured (HTTPs) communication will be available for web and HTTP-based protocols. The port for HTTP-based M2M API protocols remains the same as already set.



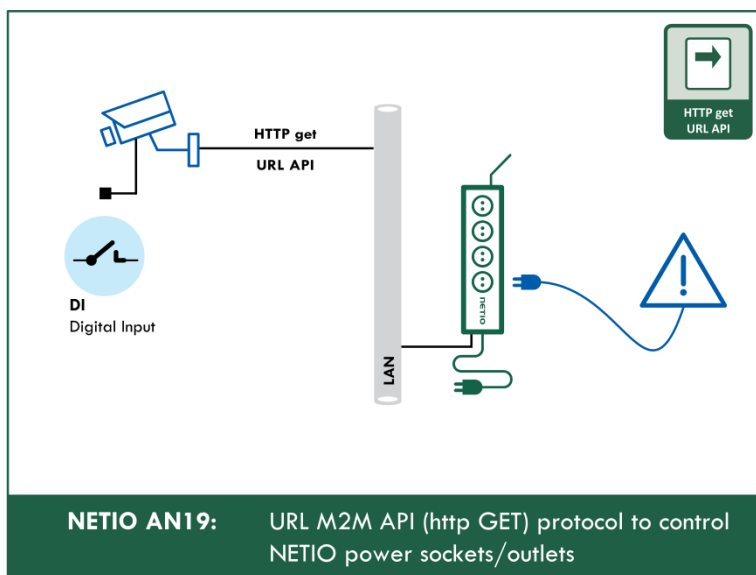
Picture 4 – Security settings GUI

- **Turn on secure connection (HTTPS)** – Enable/disable secure connection for web administration and HTTP-based M2M API protocols (JSON, XML, URL)
- **HTTPS port** – Port used for the web administration (M2M API protocols use the port set in M2M Protocol settings)
- **Allow URL-in to use insecure connection (HTTP)** – Enable unsecure connection **for URL-in - Lua only!**
- **Create New Certificate** – Generates and immediately installs a new certificate.
- **Certificate** – Info about certificate validity (certificate is generated with one year validity)
- **Fingerprint** – Certificate fingerprint

NETIO AN (Application Note)

AN19 URL API (http GET) protocol to control NETIO 4x power sockets/outlets 110 / 230V

The AN19 Application Note demonstrates the use of the “URL API” or “http get” M2M API protocol. By invoking a certain URL, one of electrical outputs 110/230V can be switched On / Off / Toggle / Short pulse. This protocol is supported by all NETIO 4x devices (NETIO 4 / 4All / 4C).



>> Read the AN19 on www.netio-products.com

Document history

Document Revision	Publication Date	Description
1.0	14.11.2017	Initial release for FW 3.0.1
1.1	18.2.2018	Few updates for FW 3.1.0
1.2	30.11.2018	AN19 description added
1.3	07.04.2021	New supported devices listed, NETIO 4C differences marked.