



wattlet
LA REVOLUTION ELECTRIQUE

Wattcube Web

API

WATTLET
719 rue Albert Camus
31190 Auterive - France

Content

1. Introduction	3
2. URL commands	3
3. Commands list	4
4. Error list	5
5. State module read	7

1. INTRODUCTION

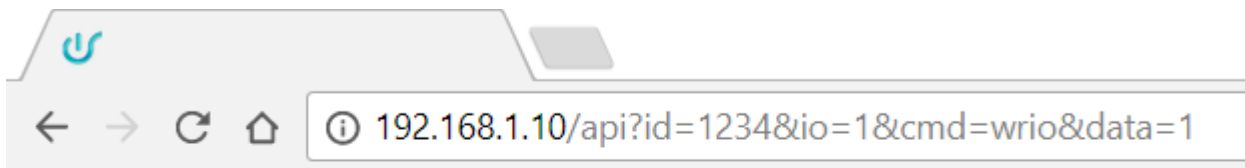
The following file describes the API interface to command a Wattcube module and get its state.

2. URL COMMANDS

URL command is a powerful way to interface the Wattcube in PLC.

A simple GET URL in any browser can command a Wattcube

Example:



```
{
  "data" : "00",
  "error" : 6
}
```

Module 001234 command to the ON state :

`http://192.168.1.10/api?id=1234&io=1&cmd=wrio&data=1`

The following command sends a broadcast command to either switch on or switch off all modules :

`http://192.168.1.10/api?id=0&io=1&cmd=wrio&data=1` to switch all on

`http://192.168.1.10/api?id=0&io=1&cmd=wrio&data=0` to switch all off

1.1.1. Syntax

Here is the command syntax:

`http://< wattcube_web_ip_address >/api?id=xxxx&io=x&cmd=xxxx&data=xx`

Parameter	Designation	Example	
wattcube_web_ip_address	Wattcube IP address (IPV4)	192.168.1.2	IP address on a local network (192.168.1.10 by default)

API

id	Address behind a Wattcube module or 000000 for a broadcast command	001234	
io		1 or 2	IO1 or IO2
cmd (command)	Write, read or programming mode command	wrio	Write command
data	Data to send to module	01	Output 1 is on

3. COMMANDS LIST

cmd (command)	Designation	Data	Answer(json)
wrio	Command output or input of a Wattcube	Write output/input: 00 => Input/output to OFF 01 => Input/output to ON For roller shutter/dimmer : 00 => Shutter/dimmer at 0% 01 => Shutter/dimmer at 12.5% ... 08 => Shutter/dimmer at 100% For Wattcube Pilot: 00 => CONFORT/COMFORT 01 => ECO 02 => HORS_GEL/FROST_FRE	<pre>{ "data" : "00", "error" : x }</pre>

API

		E 03 => ARRET/STOP 04 => COMFORT_1 05 => COMFORT_2	
rdio	Wattcube read		<pre>{ "data" : "xx", "error" : y }</pre> <p>x=0 =>input/output to OFF</p> <p>x=1 => input/output to ON</p>
prog	Programming mode	00 => End programming mode 01 => Start programming mode or programming a Wattcube that is already in programming mode 02 => Erase programming 03 => Lock programming with the Wattcube Badge 04 => Unlock programming with the Wattcube badge	<pre>{ "data" : "00", "error" : x }</pre>
eepr	Memory read		

4. QUALITY PARAMETERS

The level pictures corresponds to the following levels:

Level	SNR(dB)	RSSI(dBuV)	POWERLEVEL	IMPEDANCE	NOISE(dBuV)
-------	---------	------------	------------	-----------	-------------

API

			(dBuV)	(Ohm)	
5	25	120	120	12	80
4	20	110	110	9	90
3	15	100	100	6	100
2	10	90	90	3	110
1	5	80	80	1	120

The memory mapping is the following:

Memory address	Parameter	Designation	Ratio	Unit
0xf2	RSSI	Received signal strenght	Val= 70+x/4	dBuV
0xf3	SNR	Signal to noise ration(dB)	Val= x/4	dB
0xf4	Impedance	Impedance (ohm)	Val= x/8	Ohm
0xf5	Powerlevel	Output PLC power level	Val= 70+x/4	dBuV
0xf6	Noise	Noise on powerline	Val= 70+x/4	dBuV

5. ERROR LIST

If the error field is not 0, there is a problem.
 The error codes are as follows :

error	designation
0	NOERROR
1	SIZE
2	BUFFER_FULL
3	MEMORY_FULL
4	NOT_FOUND
5	BAD_PARAM
6	PAIRING_BUFFER_FULL
7	PLC_SSPIF_TIMEOUT
8	PLC_BYTE_RCV_TIMEOUT
9	PLC_SPEED_ERROR
10	BAD_FRAME_TYPE
11	BAD_FRAME_END

API

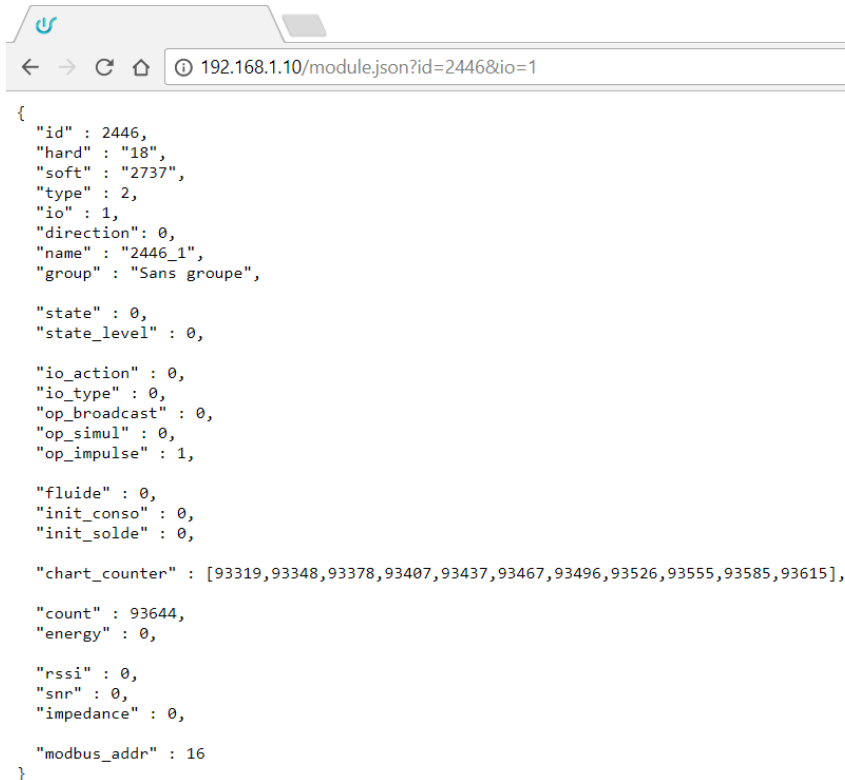
12	UNEXPECTED_FRAME_TYPE
13	BAD_CRC
14	BAD_FRAME_COUNTER
15	BAD_MODULE_TYPE
16	UART_BUSY
17	UART_OVERRUN
18	UART FRAMEERROR
19	TASK_BUSY
20	PACKET_TIMEOUT
21	PLC_RX_TIMEOUT
22	USINE_MODE
23	BAD_DATA_PACKET_HEADER
24	NO_MODULE_ANSWER
25	UART_PACKET_TIMEOUT
26	NO_MODULE
27	NOTIFICATION_PUSH
28	INVALID_PROPERTY
29	BAD_CMD
30	ID_FOUND
240	UNKNOWN_ERROR
241	BAD STATE MACHINE

6. STATE MODULE READ

The state of all known modules by the Wattcube Web can be read with a json file. Enter the following command to access to the module identity page:

<http://192.168.1.10/module.json?id=2446&io=1>

API



```

{
  "id" : 2446,
  "hard" : "18",
  "soft" : "2737",
  "type" : 2,
  "io" : 1,
  "direction": 0,
  "name" : "2446_1",
  "group" : "Sans groupe",

  "state" : 0,
  "state_level" : 0,

  "io_action" : 0,
  "io_type" : 0,
  "op_broadcast" : 0,
  "op_simul" : 0,
  "op_impulse" : 1,

  "fluide" : 0,
  "init_conso" : 0,
  "init_solde" : 0,

  "chart_counter" : [93319,93348,93378,93407,93437,93467,93496,93526,93555,93585,93615],

  "count" : 93644,
  "energy" : 0,

  "rssi" : 0,
  "snn" : 0,
  "impedance" : 0,

  "modbus_addr" : 16
}

```

Tag	Designation
id	Internal Wattcube module address (written behind any Wattcube)
hard	Wattcube module internal hardware number
soft	Wattcube module internal firmware number
type	type of wattcube (light, power, push ...)
io	io associate
direction	input/output
name	Wattcube name created by the user
group	Wattcube group created by the user
state	Wattcube state
state_level	if 1 state by level
io_action	function (standard, always on...)
io_type	type (push-button, switch...)
op_broadcast	Generalized extinction

API

op_simul	Presence simulator
op_impulse	Energy meter
fluide	fluid
init_conso	initial consumption
init_solde	initial balance
chart_counter	tab of last 11 state of counter
count	current counter
energy	current energy
rsi	rsi
snr	signal-to-noise ratio
impedance	impedance
modbus_addr	modbus address