

local http IOT API

This is a Http compliance API for controlling Ecowitt's IoT device in Json format.

1. Acquire all connected IoT device list

URL:http://192.168.4.1/get_iot_device_list or http://xx.xx.xx.xx/get_iot_device_list

where xx.xx.xx.xx is device local ip address.

Generic curl format (AC1100 and WFC01 examples see end of document):

```
curl -d 'command-string' -X POST http://IP-Address-or-Domain/parse_quick_cmd_iot
```

HTTP_GET : get_iot_device_list:

Receive:

Name	Headers	Preview	Response	Initiator	Timing
get_iot_device_list			<pre>{ "command": [{cmd: "read_quick", model: 1, id: 10576, ver: 105, rfnet_state: 1, battery: 2, signal: 4}, {cmd: "read_quick", model: 1, id: 11962, ver: 113, rfnet_state: 1, battery: 2, signal: 4}, {cmd: "read_quick", model: 2, id: 2257, ver: 101, rfnet_state: 1, battery: 9, signal: 4}] }</pre>		

```
{
"command":[
{"cmd":"read_quick","model":1,"id":10576,"ver":105,"rfnet_state":1,"battery":2,"signal":4},
{"cmd":"read_quick","model":1,"id":11962,"ver":113,"rfnet_state":1,"battery":2,"signal":4},
{"cmd":"read_quick","model":2,"id":2257,"ver":101,"rfnet_state":1,"battery":9,"signal":4}
]
} // 3 IoT devices on the hub.
or
{"command":[]} // if it is blank, then there is no device
```

Remark:

Read_quick: command type

model : 1 for WFC01, 2 for AC1100.

id : sub device ID eg: id = 10576 = 0x2950

ver : sub device firmware version

rfnet_state: 0 or 1 ; (hub to sub-device radio link) 0: currently offline, 1 : currently online

battery : 9: invalid; 0~5 battery level, when <=1 low battery

signal: 0: no signal, 0~4 signal strength indicator.

2. Turn On the Device

URL:http://192.168.4.1/parse_quick_cmd_iot/ or http://xx.xx.xx.xx/parse_quick_cmd_iot

where xx.xx.xx.xx is device local ip address.

HTTP_POST : parse_quick_cmd_iot:

1) Turn On WFC01 (0x1113) (after this command sent, read the device status once every two seconds to confirm the command execution status)

Send:

```
{"command":[{"cmd":"quick_run",
"on_type":0,"off_type":0,"always_on":1,"on_time":0,"off_time":0,"val_type":0,"val":0,"id":4371,"model":1}]
}
```

Remark:

```
"cmd":"quick_run", // command type
"on_type":0,      // fixed to 0
"off_type":0,    // fixed to 0
"always_on":1,   // always_on (fixed to 1)
"on_time":0,    // fixed to 0
"off_time":0,   // fixed to 0
"val_type":0,   // fixed to 0
"val":0,        // fixed to 0
"id":4371,      //id = 4371 =0x1113
"model":1,      // 1:WFC01, 2:AC1100。
```

2) Turn on AC1100 (0x8D1)

Send:

```
{"command":[{"cmd":"quick_run",
"on_type":0,"off_type":0,"always_on":1,"on_time":0,"off_time":0,"val_type":0,"val":0,"id":2257,"model":2}]
}
```

```
"cmd":"quick_run", // command type
"on_type":0,      // fixed to 0
"off_type":0,    // fixed to 0
"always_on":1,   // always_on fixed to 1
"on_time":0,    // fixed to 0
"off_time":0,   // fixed to 0
"val_type":0,   // fixed to 0
"val":0,        //fixed to 0
"id":2257,      // id = 2257 = 0x8D1
"model":2      // model : 1 WFC01, 2 AC1100。
```

3. Turn off the device

URL: http://192.168.4.1/parse_quick_cmd_iot/ or http://xx.xx.xx.xx/parse_quick_cmd_iot

where xx.xx.xx.xx is device local ip address.

HTTP_POST : parse_quick_cmd_iot:

Turn Off WFC01 (0x1113):

Send:

```
{"command":[{"cmd":"quick_stop","id":4371,"model":1}]}
```

Remark:

```
"cmd":"quick_stop", //quick_stop
```

```
"id":4371, // id = 4371 =0x1113
```

```
"model":1 // 1:WFC01, 2 :AC1100。
```

Turn Off the AC1100 (0x8D1)

Send:

```
{"command":[{"cmd":"quick_stop","id":2257,"model":2}]}
```

Remark:

```
"cmd":"quick_stop", //
```

```
"id":2257, // id = 2257 = 0x8D1
```

```
"model":2 // model 1:WFC01, 2: AC1100。
```

4. Acquire specified device information

URL:http://192.168.4.1/parse_quick_cmd_iot/ or http://xx.xx.xx.xx/parse_quick_cmd_iot

where xx.xx.xx.xx is device local ip address.

HTTP_POST : parse_quick_cmd_iot:

1) Read ID 4371 WFC01 device information.

Send:

```
{"command":[{"cmd":"read_device","id":4371,"model":1}]}
```

Receive:

```
{  
"command":[{"  
"model":1,"id":4371,"nickname":"WFC01-00001113","devicename":"ijtACsNMWkIKzX8rf3yQ","ver  
sion":103,"water_status":0,"warning":16,"always_on":1,"val_type":1,"val":0,"run_time":1115,"wfc01  
batt":0,"rssi":4,"gw_rssi":-20,"timeutc":1715335047,"publish_time":1715334144,"water_action":4,"wa  
ter_running":0,"plan_status":128,"water_total":10,"happen_water":10,"flow_velocity":0}]  
}
```

Remark:

```
“read_device” // command type
```

```
"model":1, // device type 1: wfc01, 2:AC1100
```

```
"id":4371, // device id = 4371 =0x1113
```

```
"nickname":"WFC01-00001113", // device name ( not changeable)
```

```
"devicename":"ijtACsNMWkIKzX8rf3yQ", // device name given on server ( not changeable)
```

```
"version":101, // version number
```

```

"water_status":0, // valve state 0: off 1: on
"warning":16, // warning bit 0: leak; bit 1: no water; bit 2: temp low; bit 3: temp high; bit 4:
low battery; bit 7: offline;
"always_on":1, // =1 always on in effective; =0 by volume or time. (current working mode)
"val_type":1, // val unit: 0:seconds 1: minutes 2: hours 3: volume (L) valid when
always_on =0
"val":0, // Quantiy ( valid when always_on =0)
"run_time":11115, // latest triggered operation running time ( seconds )
"rssi":4, // signal strength for the sub device
"gw_rssi":-24, // signal strength of the hub
"timeutc":1715324614, // hub to/from sub device data update time stamp
"publish_time":1715251415, // water_action triggered time stamp .
"water_action":35, // triggering action type ( refer to the definition listed below)
"water_running":0, // program in progress 0: completed 1: in progress
"plan_status":128, // unused
"water_total":10, // history total volume ( reset to 0 after power up)
"happen_water":0.000, // water volume when water program starts. (L = water_total - happen_water)
"flow_velocity":0.00, // flow rate (L/min) at "timeutc"
"water_temp":76.1 // water temperature (F) at "timeutc"

```

Send:

```
{"command":[{"cmd":"read_device","id":11962,"model":1}]}
```

Receive:

```

{
"command":[{"
"model":1,"id":11962,"nickname":"WFC01-00002eba","devicename":"lseK9WNPprBabxLkv0CQ","v
ersion":113,"water_status":0,"warning":144,"always_on":0,"val_type":0,"val":0,"run_time":0,"wfc01b
att":0,"rssi":4,"gw_rssi":0,"timeutc":0,"publish_time":1,"water_action":35,"water_running":0,"plan_sta
tus":128,"water_total":0.000,"happen_water":0.000,"flow_velocity":0.00"}]
}

```

2) Acquire AC1100 device info with ID x8D1(2257)

Send:

```
{"command":[{"cmd":"read_device","id":2257,"model":2}]}
```

Receive:

```

{
"command":[{"
"model":2,"id":2257,"nickname":"AC1100-000008d1","devicename":"I3WN9g6Ki7DCH0tzcZRq","v
ersion":101,"ac_status":0,"warning":0,"always_on":0,"val_type":1,"val":10,"run_time":600,"rssi":4,"g
w_rssi":-42,"timeutc":1715652373,"publish_time":1715597775,"ac_action":35,"ac_running":0,"plan_s
tatus":0,"elect_total":25,"happen_elect":0,"realtime_power":0,"ac_voltage":224,"ac_current":0}]
}

```

}

Remark:

```
"model":2, // devcie type 1: wfc01, 2:AC1100
"id":2257, // devcie id = 2257 = 0x8D1
"nickname":"AC1100-000008d1", // IOT device name
"devicename":"I3WN9g6Ki7DCH0tzcZRq", // IoT device name on the server
"version":101, // version
"ac_status":0, // relay state 0: off 1: on
"warning":0, // warning bit 0: leak current detected on off state; bit 1: no load current
detected when it is on (not in use now); bit 2: low current (not in use now); bit 3: over load ; bit 4:
relay abnormal; bit 7: not connected on radio;
"always_on":0, // =1 always on in effective; =0 by time. (current working mode)
"val_type":1, // run_time unit 0:seconds 1: minutes (valid when always_on =0)
"val":10, // QTY (valid when always_on =0)
"run_time":600, // latest triggered operation running time ( seconds )
"rssi":4, // sub device signal strength detected by the gateway.
"gw_rssi":-42, // sub device reported gateway signal strength.
"timeutc":1715652373, // hub to/from sub device data update time stamp
"publish_time":1715597775, // ac_action triggered time stamp

"ac_action":35, // triggering action type, refer to the remarks for details.
"ac_running":0, // program in progress 0: completed 1: in progress
"plan_status":0, // unused
"elect_total":25, // history total w.h ( reset to 0 after power up)
"happen_elect":0, // w.h reading when program starts. (w.h = elect_total - happen_elect)
"realtime_power":0, // power meter reading when time is "timeutc"
"ac_voltage":224, // power voltage at "timeutc"
"ac_current":0 // power current at "timeutc"
```

ACTION Triggering Source:

```
0=>'IDEL RUN' // reset state
1=>'BUTTON RUN' //button triggered run
2=>'BUTTON STOP' //button triggered stop
3=>'QUICK RUN' // Quick Run triggered by "ecowitt" APP
4=>'QUICK STOP' // Quick Run stop triggered by "ecowitt" APP
5=>'SMART RUN' // Smart triggered Run
6=>'SMART STOP' //Smart triggered Stop
7=>'PLAN RUN' // Plan triggered Run
// gateway triggered local operation
35=>'LOCAL RUN' // local WLAN triggered Run
36=>'LOCAL STOP' // local WLAN triggered Stop
```

curl call/request examples:

power on AC1100:

AC1100-1

on:

curl -d

```
'{"command":{"on_type":0,"off_type":0,"always_on":1,"on_time":0,"off_time":0,"val_type":1,"val":0,"cmd":"quick_run","id":10454,"model":2}}' -X POST
```

http://IP-address-or-domain/parse_quick_cmd_iot

off:

```
curl -d '{"command":{"cmd":"quick_stop","id":10454,"model":2}}' -X POST
```

http://IP-address-or-domain/parse_quick_cmd_iot

Status:

```
curl -d '{"command":{"cmd":"read_device","id":10454,"model":2}}' -X POST
```

http://IP-address-or-domain/parse_quick_cmd_iot

WFC01-Status:

```
curl -d '{"command":{"cmd":"read_device","id":10534,"model":1}}' -X POST
```

http://IP-address-or-domain/parse_quick_cmd_iot

start valve infinite:

curl -d

```
'{"command":{"on_type":0,"off_type":0,"always_on":1,"on_time":0,"off_time":0,"val_type":0,"val":0,"cmd":"quick_run","id":10534,"model":1}}' -X POST
```

http://IP-address-or-domain/parse_quick_cmd_iot

start valve for 1 minute:

curl -d

```
'{"command":{"on_type":0,"off_type":0,"always_on":0,"on_time":0,"off_time":0,"val_type":1,"val":1,"cmd":"quick_run","id":10534,"model":1}}' -X POST
```

http://IP-address-or-domain/parse_quick_cmd_iot

off:

```
curl -d '{"command":{"cmd":"quick_stop","id":10534,"model":1}}' -X POST
```

http://IP-address-or-domain/parse_quick_cmd_iot