Colorlight

Cx Async Controler RESTful API

- Colorlight interconnection

TABLE OF CONTENTS

1. Get Device Generic Info

http://192.168.42.129/api/info.json

Method:GET

- The IP address "192.168.42.129" is the target Cx Led Player ip address here as an example.
- This API will return JSON as following:

```
{
  "info": {
    "vername": "1.64.6",
    "serialno": "CLCC4000A008",
    "model": "c4",
    "up": 9989856,
    "mem": {
      "total": 1073741824,
      "free": 778567680
    },
    "storage": {
      "total": 5878841344,
      "free": 5878644736
    },
    "playing": {
      "name": "new.vsn",
      "path": "/mnt/sdcard/Android/data/com.color.home/files/Ftp/program",
      "source": "lan"
    }
  }
```

- info.vername:Version number of the device
- info.up:Device UP time (miliseconds).
- storage.total: Total internal storage size. (Bytes)
- storage.free: Remaining internal storage size available. (Bytes)
- playing.name: The program name that is presently playing on the LED.
- Playing.source: Current playing program source. The program source types are as following:
  - lan: The program is published from local network, including WIFI/LAN/USB cable.
  - usb: The program is in external USB storage.
  - usb-synced: The program is synchronized from (copied from) external USB storage into

internal storage.

- internet: The program comes from Internet.

2. Screenshot

http://192.168.42.129/api/screenshot

Method:GET

Returns current screenshot in PNG format.

3. Enable/Disable Toast

http://192.168.42.129/api/ showtoast

Method:POST

Content-type:application/json; charset=utf-8

Body

```
{
   "showToast": 0|1
}
```

Description

Whether to display the playing program source and name on program start.

NOTE: You have to reboot the device after switching this show toast flag, otherwise, it won't take effect.

4. Get Toast Status

http://192.168.42.129/api/ showtoast.json

Method:GET

Return

```
{
   "showProgramToast": 1
}
```

Description

Check the toast flag, whether currently the device will display program source and name on switching the program.
NOTE: You have to reboot the device after switching this show toast flag using "Enable/Disable Toast" API, otherwise, it won't take effect.

5. Get Programs

http://192.168.42.129/api/vsns.json

Method:GET

Retrieve all the programs in the device:

```
{
  "playing": {
    "type": "lan",
    "name": "new.vsn"
  },
  "contents": [
    {
      "type": "lan",
      "content": [
    {
      "name": "12345.vsn"
      "size": 7665246,
      "md5": "882024f3d5869aad992a58fec123a19c"
      "publishedmd5": "8B2E1E8BE7588F7862A47DA9D7C7F670"
    },
    {
      "name": "256256.vsn"
      "size": 7665246,
      "md5": "882024f3d5869aad992a58fec123a19c"
      "publishedmd5": "8B2E1E8BE7588F7862A47DA9D7C7F670"
    }
  ]
},
{
  "type": "usb-synced",
  "content": [
    {
      "name": "new.vsn",
      "size": 7665246,
      "md5": "882024f3d5869aad992a58fec123a19c"
      "publishedmd5": "8B2E1E8BE7588F7862A47DA9D7C7F670"
    }
  ]
},
{
  "type": "usb",
  "content": [
    {
      "name": "new.vsn",
      "size": 7665246,
      "md5": "882024f3d5869aad992a58fec123a19c"
      "publishedmd5": "8B2E1E8BE7588F7862A47DA9D7C7F670"
    }
  ]
},
{
  "type": "lan",
  "content": [
    {
      "name": "new.vsn",
      "size": 11454370,
      "md5": "eb896b2c17d5638f7fbd18db7d3e0c4"
      "publishedmd5": "8B2E1E8BE7588F7862A47DA9D7C7F670"
```

```
        }
      ]
    }
  ]
}
```
playing.type: The source of the playing program.

The program source types are as following:

- lan: The program is published from local network, including WIFI/LAN/USB cable.
- usb: The program is in external USB storage.
- usb-synced: The program is synchronized from (copied from) external USB storage into internal storage.
- internet: The program comes from the Internet.

playing.name: The program that is currently playing on the LED.
Contents: Array of 4 types program source.
Contents[x].type: Program source.
Contents[x].content[i].name: Program name.
Md5: Program md5.
Size: The total size of the program, including the assets size and the program meta data.

6. Switch Program

http://192.168.42.129/api/vsns/sources/lan/vsns/new.vsn/activated

Method: PUT

Content-type:application/json; charset=utf-8

- The "lan" in the example is the program source.
  All program source types available are as following:
  - lan: The program is published from local network, including WIFI/LAN/USB cable.
  - usb: The program is in external USB storage.
  - usb-synced: The program is synchronized from (copied from) external USB storage into internal storage.
  - internet: The program comes from Internet.
- The "new.vsn" is the program name, which you'd like to switch to.

  * If you publish a program from the USB cable, the source is "lan".

7. Delete Program

http://192.168.42.129/api/vsns/sources/lan/vsns/q.vsn

Method:DELETE

Content-type:application/json; charset=utf-8

- The "lan" in the example is the program source.
  All program source types available are as following:
  - lan: The program is published from local network, including WIFI/LAN/USB cable.
  - usb: The program is in external USB storage.
  - usb-synced: The program is synchronized from (copied from) external USB storage into internal storage.
  - internet: The program comes from Internet.
- The "q.vsn" is the program name, which you'd like to remove from the device.

  * If you publish a program from the USB cable, the source is "lan".

8. Get Network Status

http://192.168.42.129/api/ifstatus.json

Method:GET

This interface return current network interfaces status.
The following is an example of the returned network interface status:

```
{
        "types": [
                {
                        "peers": [
                                {
                        "ip": "192.168.43.159",
                        "mac": "d8:1d:72:49:cc:ae"
                                }
                        ],
                "SSID": "C3L1",
        "pass": "123456789",
        "type": "wifi ap",
        "operstate": "up",
        "ips": {
            "broadcast": "192.168.43.255",
            "ip": "192.168.43.1",
            "mask": "255.255.255.0"
        },
        "mac": "a8:ab:60:00:00:03",
        "enabled": 1,
        "connected": 0,
        "carrier": 1
        "channel":1
    },
    {
        "currentap": "dd-wrt--07",
        "state": "COMPLETED",
        "ssids": [
            {
                "SSID": "\"cltap\"",
                "pass": "*",
                "priority": 0
            },
            {
                "SSID": "\"GSA-NG5G\"",
                "pass": "*",
                "priority": 0
            },
            {
                "SSID": "\"gsa-TP5G\"",
                "pass": "*",
                "priority": 0
            },
            {
                "SSID": "\"dd-wrt-07\"",
                "pass": "*",
                "priority": 0
            },
            {
                "SSID": "\"dd-wrt--07\"",
                "pass": "*",
                "priority": 0
            },
            {
                    "SSID": "\"dd-wrt--0\"",
                    "pass": "*",
                    "priority": 0
                }
            ],
            "speed": 54,
            "type": "wifi",
            "operstate": "up",
            "ips": {
                "broadcast": "192.168.7.255",
                "ip": "192.168.7.134",
                "mask": "255.255.255.0"
            },
            "mac": "a8:ab:60:00:00:03",
            "enabled": 1,
            "connected": 0,
            "carrier": 1
        },
        {
            "type": "lan",
            "operstate": "up",
            "mode": "dhcp",
            "ips": {
                "dns1": "192.168.7.1",
                "dns2": "",
                "gateway": "192.168.7.1",
                "ip": "192.168.42.129",
                "mask": "255.255.255.0"
            },
            "mac": "a8:aa:60:00:00:03",
            "enabled": 1,
            "connected": 0,
            "carrier": 1
        }
        {
            "type": "4G",
            "enabled": 1,（4G enable status,0 switched off,1 switched on）
            "strength": 1,（Signal strenth,1 Very weak,2 Weak,3 Medium,4 Strong）
            "mode": "CDMA",（Network mode: GSM|HSDPA|WCDMA|CDMA|LTE, etc.）
            "log": 1（Wether to log the dial,0-donot log,1-log）
            "connected": 0,
        }
    ]
}
```

- types[x].type: network types, contains:
  a) types[0].type is "wifi ap",
  b) types[1].type is "wifi",
  c) types[2].type is "lan",
  d) types[3].type is "4G"

1. Wifi ap
   - types[0] – "wifi ap" status. Data will be valid if current device is acting as WIFI AP
   - types[0].peers[x]: Devices currently connecting to the Async LED Player.
   - types[0].peers[x].ip: the ip address of the device currently connecting to the Async LED Player.
   - types[0].peers[x].mac: the MAC address of the device currently connecting to Async the LED Player.
   - types[0].SSID: the Async LED Player Wifi ap SSID,
   - types[0].pass: the Async LED Player Wifi ap password,
   - types[0].ips: the Async LED Player Wifi ap ip info
   - types[0].mac: the Async LED Player Wifi ap mac
   - types[0].enabled:Whether the Async LED Player is acting as an Wifi ap
2. wifi
   - types[1].state:current wifi state:
     - COMPLETED
     - DISCONNECTED
     - SCANNING
   - types[1].currentap: the wifi ap currently connected
   - types[1].speed: connection speed (Mbps)
   - types[1].type: wifi
   - types[1].ips: the device's ip address
   - types[1].mac: the device's mac address
3. lan
   - types[2].type: lan
   - types[2].mode: "dhcp" or "static"
   - types[2].ips: ip info
   - types[2].mac: lan mac address

   * when there is no lan cable plugin, the "carrier" will be 0.
4. 4g
   - "type": "4g",
   - "enabled": 0|1,（4G enable status,0 switched off,1 switched on）
   - "strength": 1|2|3|4,（Signal strength,1 Very weak,2 Weak,3 Medium,4 Strong）
   - "mode": "LTE",（Network modes: GSM|HSDPA|WCDMA|CDMA|LTE, etc.）
   - "log": 0|1（Whether to log the dial,0-donot log,1-log）
   - "connected": 0|1, connected or not.

9. Configure Network

http://192.168.42.129/api/network.json

Method:GET

Retrieve the device's current network configuration.

```
{
        "types": [
                {
                        "SSID": "dd-wrt--07",（the wifi ap this device is connecting to）
                        "ips": {},
                        "pass": "180380580",（wifi pass）
                        "type": "wifi",
                        "enabled": 0,（wifi switch,0-switched off,1-switched on）
                        "isstatic": 0
                },
                {
                        "dns1": "",
                        "dns2": "",
                        "ips": {
                                "gateway": "",
                                "ip": "192.168.1.1",（ip）
                                "mask": "255.255.255.0"（netmask）
                        },
                        "type": "lan",
                        "enabled": 1,（lan switch,0-off,1-on）
                        "isstatic": 1（static ip or not, 1-static ip, 0-dhcp）
                },
                {
                        "SSID": "C6" ,(device's ap name)
                        "ips": {},
                        "pass": "123456789", (device' ap password)
                        "type": "wifi ap",
                        "channel":1
                        "enabled": 1,（wifi enable status, 0-off,1-on）
                        "isstatic": 0
                }
                {
                    "mode": "LTE", （mobile network mode）
                    "type": "4G",
                    "enabled": 1, （enabled or not）
                    "log": 0, （log or not）
                    "strength": 3（Signal strength,1: very weak, 2:weak, 3:medium, 4:strong）
                }
        ]
  }
```

* ONLY one of the "wifi" | "4g" | "lan" can be enable at the same time. For instance if the "wifi" is enabled, DONT enable the "lan" and vice versa.
* Before set the network, please retrieve firstly the network.json info, and change the configuration base on the retrieved network.json.

10. Device Power Management

http://192.168.42.129/api/action

Method:POST

Content-type:application/json; charset=utf-8

HTTP Body:

Sleep

{"command":"sleep"}

Wakeup

{"command":"wakeup"}

Reboot

{"command":"reboot"}

11. Configure Device Meta Data

PUT http://192.168.42.129/api/terminal

Method PUT

Content-type:application/json; charset=utf-8

HTTP BODY

```
{
   "name": "Terminal00A1",
   "leddescription": "Colorlight's test terminal"
}
```

Name: device name to be set
Leddescription: device description

12. Get Device Meta Data

http://192.168.42.129/api/terminal.json

Method GET

Content-type:application/json; charset=utf-8

HTTP BODY

```
{
   "name": "Terminal00A1",
   "leddescription": "Colorlight's test terminal"
}
```

Name:device name
Leddescription:device description

13. Get Device Power Status

http://192.168.42.129/api/powerstatus.json

Method GET

Content-type:application/json; charset=utf-8

HTTP BODY

{"powerstatus": 0}
Status:Device power status 0--sleep,1--wakeup

14. Get Device Time

http://192.168.42.129/api/rtc.json

Method GET

Content-type:application/json; charset=utf-8

HTTP BODY

{"time": "2016-3-25 17:39:22","timezone":8,"isautotimezone":1,"isautotime":1}
time :device time
timezone:timezone
isautotimezone:auto timezone or not,1-auto,0-manual
isautotime:auto NTP time or not, 1-auto,0-manual

15. Configure Time

http://192.168.42.129/api/rtc

Method:PUT

Content-type:application/json; charset=utf-8

HTTP BODY:

{"time": "2016-3-25 17:39:22","timezone":8,"isautotimezone":1,"isautotime":1}
time : device time to be set
timezone: timezone
isautotimezone: auto timezone or not,1-auto,0-manual
isautotime: auto NTP time or not, 1-auto,0-manual

16. Configure Locale

http://192.168.42.129/api/locale

Method:PUT

Content-type:application/json; charset=utf-8

HTTP BODY:

{"language ": "zh","country": "CN"}
language :
https://en.wikipedia.org/wiki/List_of_ISO_639-1_codes

country:
https://en.wikipedia.org/wiki/ISO_3166-2
* NOTE: the device is going to reboot automatically once the resolution is changed.

17. Retrieve Locale

GET http://192.168.42.129/api/locale.json

Method GET

Content-type:application/json; charset=utf-8

HTTP BODY:

{"language ": "zh","country": "CN"}

language :
https://en.wikipedia.org/wiki/List_of_ISO_639-1_codes

country:
https://en.wikipedia.org/wiki/ISO_3166-2

18. Configure LED Resolution

PUT http://192.168.42.129/api/dimension

Content-type: application/json;charset=utf-8

HTTP BODY 如下:

{"width":128,"height":256, "hsync":0,"dclk": 0}

width: led width (MUST be multiple of 16, max 4096)
height: led height (MUST be multiple of 64,max1536)
hsync: Must be 0 for the default 60FPS LED refreshing rate.
dclk: Must be 0 for the default 60FPS LED refreshing rate.

* NOTE: the device is going to reboot automatically once the resolution is changed.

19. Retrieve Led Resolution

GET http://192.168.42.129/api/dimension.json

Content-type: application/json;charset=utf-8

HTTP BODY:

```
{
   "dclk": 33000000,
   "fps": 109,
   "height": 128,
   "hsync": 1647,
   "real_dclk": 33000000,
   "real_height": 128,
   "real_width": 256,
   "width": 256
}
```

width:width configured
height:height configured
hsync: hsync configured or auto calculated
dclk: dclk configured or auto calculated
fps:Frame per sec
real_width: current effective LED width
real_height: current effective LED height
real_dclk: current effective LED dclk

20. Quick Send Single-Line Text Program

POST http://192.168.42.129/api/program/singletext

Content-type: application/json;charset=utf-8

HTTP BODY:

```
{
    "text":"Single line text, hello!", (Defautlt: "")
    "x":0,( x coordination)
    "y":0,( y coordination)
    "width":256,(single line text window width)
    "height":256,( single line text window height)
    "font": {
        "name":"隶书|楷体|黑体|宋体|仿宋|default", (For English please use default)
        "size":8,    (Font size, Default: 24)
        "style":{
            "i":0|1,
            "b":0|1,
            "u":0|1
        }

        "color":"0xFFBBAABB", (Default: 0xFFFF0000)
    }
    "bgcolor":"0xAAAAAAAA" (Default: 0xFF000001)
    "scroll": {
        "dir":"left ", (Default: left)
        "isconnected":0|1, (Default: 0)
        "speed":60 (Default: 60pixel/sec)
    }(* If no scroll need, don't use the "scroll" key.)
}
```

- Text: text characters to be displayed
- Font: font properties {

  Name: font name

  Size:font size

  Style:font style{

  I:Italic

  B:Bold

  U:Underline

  }

  Color:text color, 0x "alpha""red""green""blue" in hex

}

Bgcolor: background color. 0x "alpha""red""green""blue" in hex

Scroll: scroll or not {

Dir:"left"

isConnected:0|1 is text tail connected to the head to form an infinity looping scroll text.

Speed: speed in pixel/sec.

}

CURL Example

curl -X POST -H "Content-Type:application/json" -d "{\"text\":\"Welcome to Colorlight CPlayer RESTful API!\",\"x\":0,\"y\":0,\"width\":256,\"height\":128,\"font\":{\"name\":\" 宋体\",\"size\":72,\"style\":{\"i\":0,\"b\":0,\"u\":0},\"color\":\"0xFFFF0000\"},\"bgcolor\":\"0xFF000001\",\"scroll\":{\"dir\":\"left\",\"isconnected\":0,\"speed\":60}}" http://192.168.42.129/api/program/singletext

NOTE:192.168.42.129 is the ip address when the PC is connecting with the device "CONFIG" port via USB cable bundled。

21. Configure Sending Card(Dimension, Control Area)

PUT http://192.168.42.129/api/sendingcard

Content-type: application/json;charset=utf-8

HTTP BODY 如下:

```
{
    "netareas":[ （LED DISPLAY ports control area）
        {
            "width":1024,
            "height":1024,
            "startx":10,
            "starty":30
        },
        {
            "width":256,
            "height":512,
            "startx":70,
            "starty":80
        }
    ]
}
```

22. Retrieve Sending Card info

GET http://192.168.42.129/api/sendingcard.json

Content-type: application/json;charset=utf-8

HTTP BODY:

```
    {
        "netareas":[
            {
                "width":1024,
                "height":1024,
                "startx":10,
                "starty":30
            },
            {
                "width":256,
                "height":512,
                "startx":70,
                "starty":80
            }
        ]
    }
```

23. Get Sensor Data (C6 ONLY)

GET http://192.168.42.129/api/sensor.json

Content-type: application/json;charset=utf-8

HTTP BODY 如下:

```
{
    "whichPort": "A|B", (Serial port no.)
    "brightness": 0,
        "hasAcousticSensor": true,（Noice sensor）
        "hasBrightnessSensor": false,
        "hasNoise": 0|1,
        "hasSmoke": 0|1,
        "hasSmokeSensor": true,
        "hasTempAndHumSensor": true,
        "hasTemperatureSensor": true,
        "humidity": 33,
        "temperature": 1.248
        "timeStamp": "Tue May 24 14:39:55 GMT+08:00 2016"
}
```

24. Retrieve Volume Level

GET http://192.168.42.129/api/volume.json

Content-type: application/json;charset=utf-8

HTTP BODY:

```
{
        "musicvolume":10
}
```

The musicvolume value between 0 to 15.

25. Configure Volume Level

PUT http://192.168.42.129/api/volume

Content-type: application/json;charset=utf-8

HTTP BODY:

```
{
        "musicvolume":10(0-15)
}
```

26. Switch Sending Card Input Mode(C6 ONLY)

PUT http://192.168.42.129/api/inputmode

Content-type: application/json;charset=utf-8

HTTP BODY:
{"inputmode":"dvi"} Prioritize the async LED player signal.
{"inputmode":"hdmi"} Prioritize to the HDM input signal.

27. Retrieve Sending card input mode

GET http://192.168.42.129/api/inputmode.json

Content-type: application/json;charset=utf-8

HTTP BODY:

```
{
  "inputmode": "hdmi",   // the prior input mode selected
  "inputmodeactive": "dvi" // actual input mode
}
```

28. PING IP (Domain)

POST http://192.168.42.129/api/ping

Content-type: application/json;charset=utf-8

Request HTTP Body:
{"ping":"www.lednets.com"}

Response HTTP Body:
{"resptime":2.01} miliseconds, -1 on network issue.

29. Set FPS （Frame rate）

PUT http://192.168.42.129/api/fps

Content-type: application/json;charset=utf-8

{"fps":60}

30. Retrieve FPS （Frame rate）

GET http://192.168.42.129/api/fps.json

Content-type: application/json;charset=utf-8

HTTP BODY:
{"fps":60}

31. Clean Programs

DELETE http://192.168.42.129/api/clrprgms

Clear all programs.

32. Clean Program Assets Cache

DELETE http://192.168.42.129/api/clrcache

33. Clean Cache

DELETE http://192.168.42.129/api/ clrresunused

34. Quick Sending Any Text Based Program

POST **http://192.168.42.129/api/program/program_name.vsn**

1) **Method**:POST
2) **Content-type: application/json;charset=utf-8**
3) URL: http://your_cx_async_controller_domainname_or_ip/api/program/program_name.vsn
4) Program Name: has .vsn extension, and is the last path in the API URL: http://your_cx_async_controller_domainname_or_ip/api/program/program_name.vsn
5) The "Program_name" will be created in the Async controller, and played back instantly.

HTTP Body example:

```
{
  "Programs": {
    "Program": {
      "Pages": [{
          "Regions": [
              {
                "Rect": {
                  "X": "0",
                  "Y": "0",
                  "Width": "128",
                  "Height": "64"
                },
                "Items": [{
                    "Type": "5",
        "Text":"Test Multiple line scroll text",
                    "IsScroll": "1"
                }]
            },
            {
                "Rect": {
                    "X": "0",
                    "Y": "64",
                    "Width": "128",
                    "Height": "64"
                },
                "Items": [{
                    "Type": "5",
                    "Text": "Test Multiple line Multiple page",
                    "LogFont": {
                        "lfHeight": "36",
                        "lfWidth": "0"
                    }
                }]
            }
        ]
    }]
  }
 }
}
```

Curl Example

```
curl -X POST -d @docApi.json -H "Content-Type:application/json;charset=UTF-8" "http://192.168.42.129/api/program/test9.vsn"
```

Please check the Demo/ folder for the detailed example.
Please check the "Colorlight program JSONSpec-v1.x.pdf" for the detailed Colorlight Program in JSON format for POSTing any Text based program to the Colorlight Cx Asynchronous controller.

35. Sending Any Program

Since V1.32, the Colorlight Cx Asynchronous controller supports uploading and instantly play any supported Program via HTTP POST.

POST http://192.168.42.129/api/program/program_name.vsn

Method:POST

Content-type: multipart/form-data

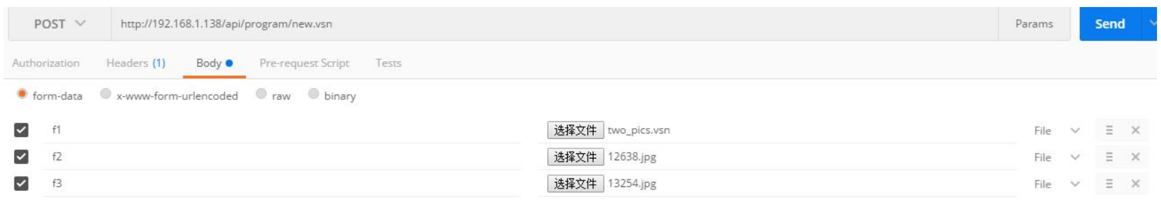
1) The Colorlight Program description file must align with the Program Spec, and be attached to the HTTP POST as an .vsn file. Please reference to "Colorlight Program (VSN) Format-v1.2.docx" for the JSON format.
2) All the assets (pictures/video/text, etc.) must also be attached to the HTTP POST in the multipart/form-data.

Curl E.g.

```
curl -F "f1=@two_pics.vsn" -F "f2=@assets/12638.jpg" -F"f3=@assets/13254.jpg" http://192.168.42.129/api/program/two_pics.vsn
```

Please check the curl demo under folde Demo/Demo_SendAnyProgram for the detailed info.

POSTMAN E.g.



http://192.168.1.138/api/program/new.vsn

● form-data ○ x-www-form-urlencoded ○ raw ○ binary

✓	f1	选择文件 two_pics.vsn	File
✓	f2	选择文件 12638.jpg	File
✓	f3	选择文件 13254.jpg	File

36. Configure Brightness

PUT http://192.168.42.129/api/brightness

Method PUT

Content-type:application/json; charset=utf-8

HTTP BODY

```
{"brightness":60}
```

brightness: An integer between 0 to 255.

37. Configure color temperature

PUT http://192.168.42.129/api/colortemp

Method PUT

Content-type:application/json; charset=utf-8

HTTP BODY

```
{"colortemp":60}
```

colortemp: An integer between 2000 to 10000.

38. Retrieve color temperature and brightness

GET

http://192.168.42.129/api/brightnessandcolortemp.json

Method GET

HTTP BODY

```
{
        "brightness":60,
        "colortemp":6000
}
```

brightness: An integer between 0 to 255.
colortemp: An integer between 2000 to 10000.

39. Save brightness and color temperature

PUT

http://192.168.42.129/api/savebrightnessandcolortemp

40. Set NTP

PUT http://192.168.42.129/api/ntp

Content-type:application/json; charset=utf-8

HTTP BODY

```
{
"ntpserver":"192.168.1.33",
"ntpinterval":3600,
"ntpthreshold":5000
}
```

ntpserver:NTP Server ip address/domain name.
ntpinterval: NTPpolling interval.
ntpthreshold: NTP time difference threshold.

41. Get NTP settings

GET http://192.168.42.129/api/ntp.json

Content-type:application/json; charset=utf-8

HTTP BODY

```
{
"ntpserver":"192.168.1.33",
"ntpinterval":3600,
"ntpthreshold":5000
}
```

42. Configure Terminal Account on Cloud Server

PUT http://192.168.42.129/api/account

Content-type: **application/json; charset=utf-8**

HTTP BODY

```
{
   "name": "terminal_account_name",
   "password": "password_configured_in_server ",
   "url": "https://your_cloudserver.com"
}
```

name: Terminal account name which was configured in the Server when add a new Terminal.
password: Terminal account password which was configured in the Server when add a new Terminal.
url: Server URL

43. Retrieve Terminal Account

GET http://192.168.42.129/api/account.json

Content-type: **application/json;charset=utf-8**

HTTP BODY

```
{
   "name": "terminal_account_name",
   "password": "password_configured_in_server",
   "url": "https://your_cloudserver.com",
      "internet" : "true|false",
   "devicestatus" : "wakeup|sleep",
   "login":"true|false"
```

}

name: Terminal account name
password: Terminal account password
url: Server URL
internet: Is internet available
login: Is login OK

44. Retrieve the Data of External Sensors （485 Sensor interface）

GET http://192.168.42.129/api/csensor.json

Content-type: application/json;charset=utf-8

Response parameters：

Parameter	Type	Example
pm25	int	30
pm10	int	50
brightness	int	0~46000
humidity	float	13.1
smoke	int	
noise	float	
temperature	float	23.1

Return example:

```
{
   "brightness":365,
   "humidity": 71.3,
   "temperature": 27.3,
   "pm25" : 30
   }
```

Description: The return value is {} which means no sensor is connected.

45. Upgrade Device’s Version

POST http://192.168.42.129/api/upgrade

Content-type: application/json;charset=utf-8

HTTP BODY

```
{
  "source": "lan"
}
```

- The "source" in the example is the upgrade file source.
  All source types available are as following:
  - lan: upload update.zip to ftp://192.168.42.129/update/ .
  - usb: put update.zip to U-Disk/update/ .
  - internet: it will auto upgrade after downloaded update.zip from C-Cloud.