



Remote Management Card

RMCARD205

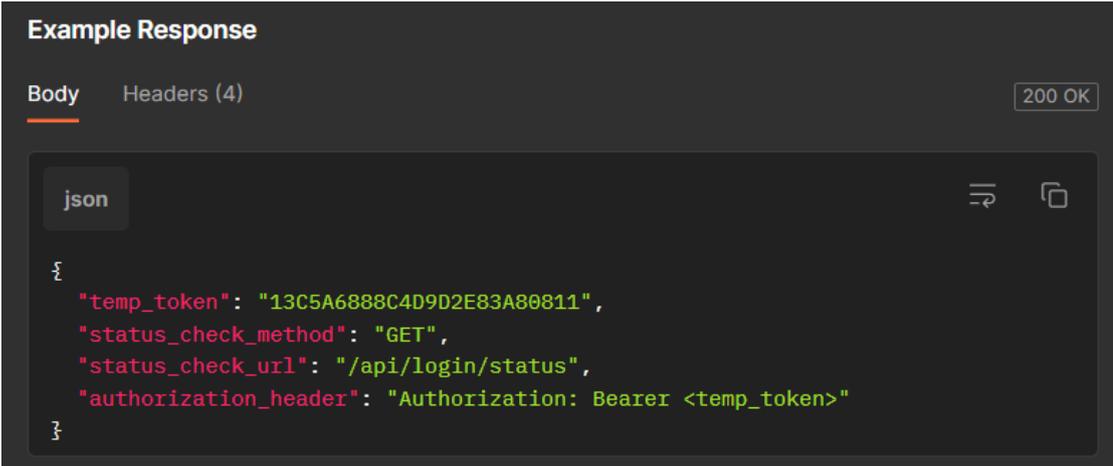
REST API Manual

The Remote Management Card allows a UPS system and environmental sensor to be managed, monitored, and configured.

1. Login

- **Step 1:** This step involves sending a POST request with login credentials to obtain a temporary token, which is then used to verify the login status and authorize subsequent requests.

```
curl -X POST https://<IP>/api/login/ -d  
'{"username":"<Account>","passwd":"<Password>"}
```



Example Response

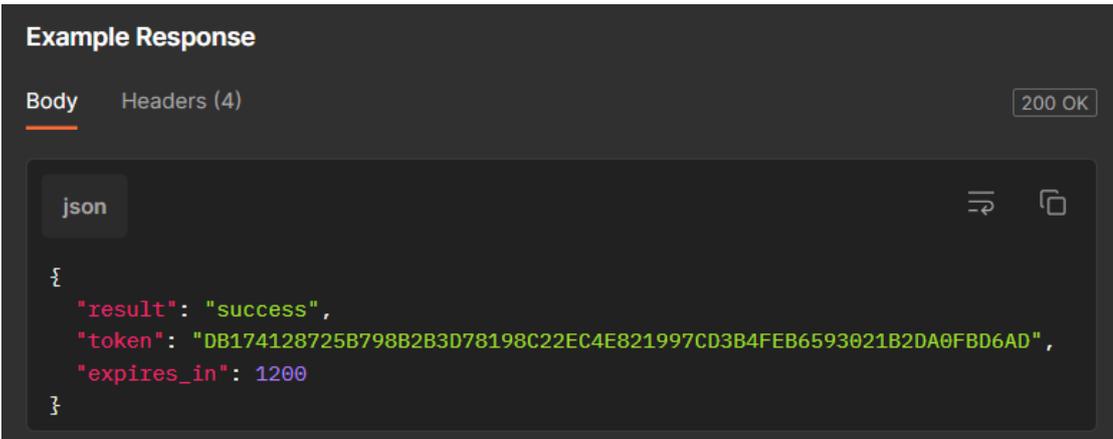
Body Headers (4) 200 OK

```
json
```

```
{  
  "temp_token": "13C5A6888C4D9D2E83A80811",  
  "status_check_method": "GET",  
  "status_check_url": "/api/login/status",  
  "authorization_header": "Authorization: Bearer <temp_token>"  
}
```

- **Step 2:** This step verifies the user's login status by sending a GET request with the Authorization header, which contains the `<temp_token>` obtained in Step 1.

```
curl -v https://<IP>/api/login/status/ -H "Authorization: Bearer  
<temp_token>"
```



Example Response

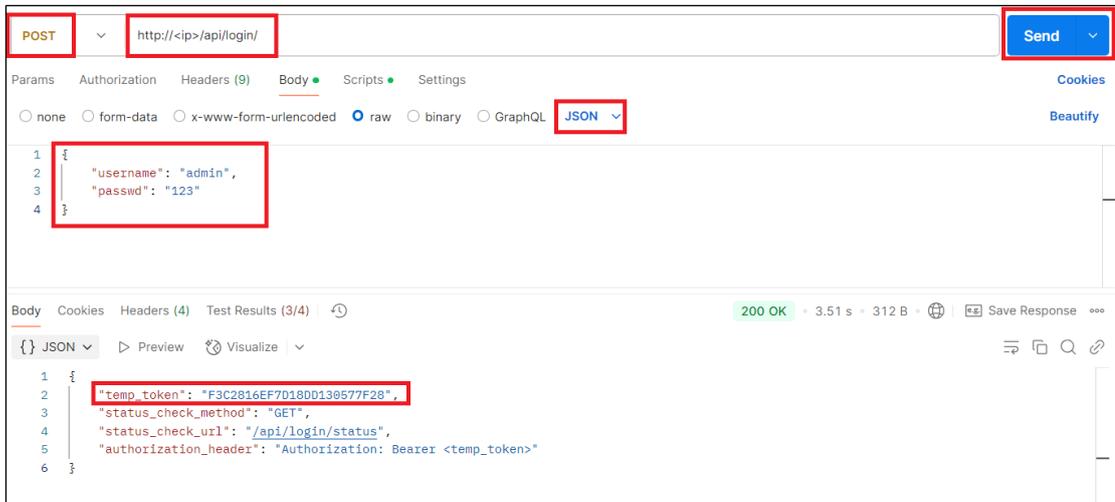
Body Headers (4) 200 OK

```
json
```

```
{  
  "result": "success",  
  "token": "DB174128725B798B2B3D78198C22EC4E821997CD3B4FEB6593021B2DA0FBD6AD",  
  "expires_in": 1200  
}
```

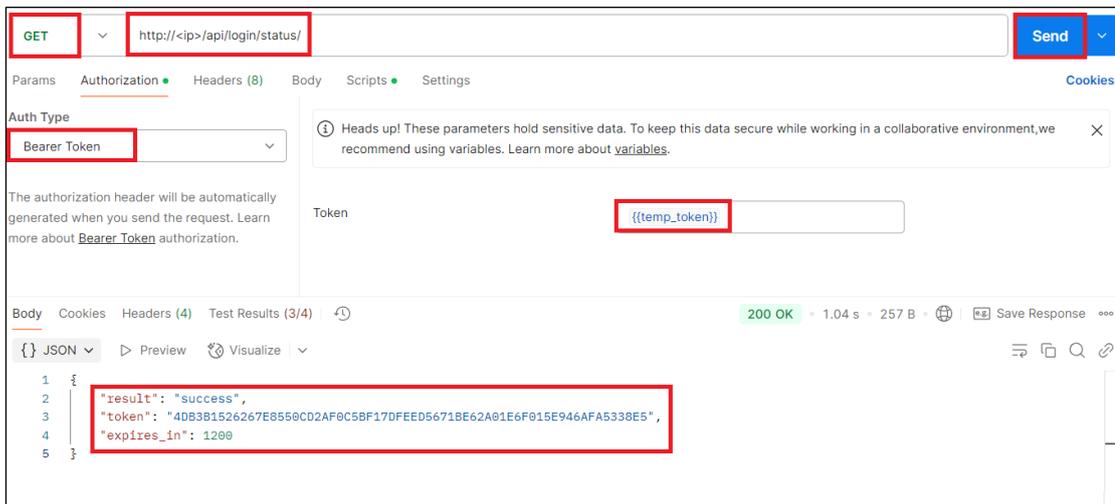
Example for POSTMAN

● Step 1:



The screenshot shows a Postman interface for a POST request to `http://<ip>/api/login/`. The request body is a JSON object: `{ "username": "admin", "passwd": "123" }`. The response is a 200 OK status with a JSON body: `{ "temp_token": "F3C2816EF7D18DD138577F28", "status_check_method": "GET", "status_check_url": "/api/login/status", "authorization_header": "Authorization: Bearer <temp_token>" }`.

● Step 2:



The screenshot shows a Postman interface for a GET request to `http://<ip>/api/login/status/`. The authorization type is set to Bearer Token, and the token field contains `{{temp_token}}`. The response is a 200 OK status with a JSON body: `{ "result": "success", "token": "40B3B1526267E8558CD2AF0C5BF17DFEED5671BE62A01E6F015E946AFA5338E5", "expires_in": 1200 }`.

2. Logout

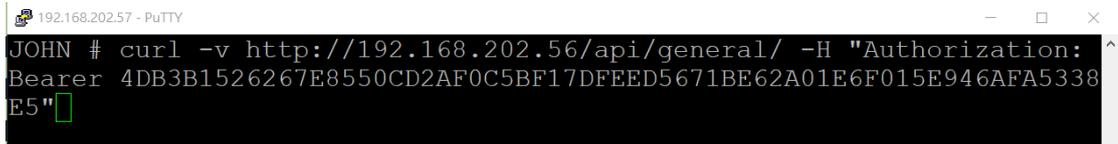
```
curl -X PUT https://<IP>/api/logout/ -d '{"logout":"true"}' -H "Authorization: Bearer <token>"
```

3. Force change different username/password

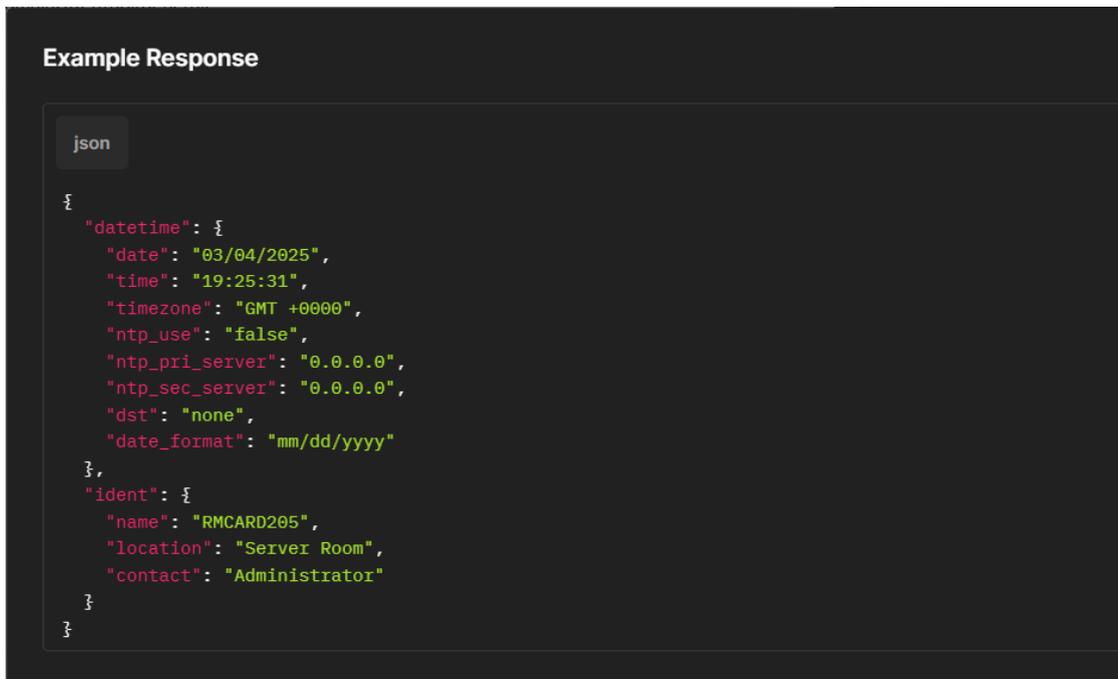
```
curl -X PUT https://<IP>/api/login/update/default_creds/ -d '{"new_username": "admin", "new_password": "123", "confirm_password": "123"}' -H "Authorization: Bearer <token>"
```

4. General

```
curl -v https://<IP>/api/general/ -H "Authorization: Bearer <token>"
```



A terminal window titled "192.168.202.57 - PuTTY" showing the execution of a curl command. The command is: `curl -v http://192.168.202.56/api/general/ -H "Authorization: Bearer 4DB3B1526267E8550CD2AF0C5BF17DFEED5671BE62A01E6F015E946AFA5338E5"`. The output shows the command being executed and the response headers.

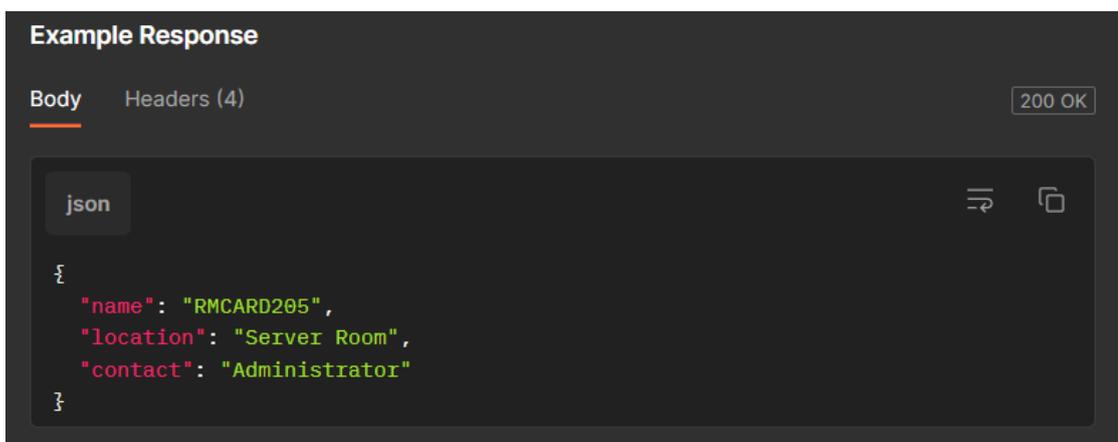


An API response viewer showing an example response. The response is in JSON format and contains the following data:

```
{
  "datetime": {
    "date": "03/04/2025",
    "time": "19:25:31",
    "timezone": "GMT +0000",
    "ntp_use": "false",
    "ntp_pri_server": "0.0.0.0",
    "ntp_sec_server": "0.0.0.0",
    "dst": "none",
    "date_format": "mm/dd/yyyy"
  },
  "ident": {
    "name": "RMCARD205",
    "location": "Server Room",
    "contact": "Administrator"
  }
}
```

- show ident

```
curl -v https://<IP>/api/general/ident/ -H "Authorization: Bearer <token>"
```



An API response viewer showing an example response for the /ident/ endpoint. The response is in JSON format and contains the following data:

```
{
  "name": "RMCARD205",
  "location": "Server Room",
  "contact": "Administrator"
}
```

- set identification name

```
curl -X PUT https://<IP>/api/general/ident/name/ -d '{"name": "<NAME>"}' -H
```

"Authorization: Bearer <token>"

The rest follows the same pattern

curl -v https://<IP>/api/general/datetime/ -H "Authorization: Bearer <token>"

curl -v https://<IP>/api/general/datetime/date/ -H "Authorization: Bearer <token>"

curl -X PUT https://<IP>/api/general/datetime/date/ -H "Authorization: Bearer <token>"

{"date": "mm/dd/yyyy"} → Ex : {"date": "11/30/2023"}

curl -v https://<IP>/api/general/datetime/time/ -H "Authorization: Bearer <token>"

curl -X PUT https://<IP>/api/general/datetime/time/ -H "Authorization: Bearer <token>"

{"time": "hh:mm:ss"} → Ex : {"time": "14:32:45"}

curl -v https://<IP>/api/general/datetime/timezone/ -H "Authorization: Bearer <token>"

curl -X PUT https://<IP>/api/general/datetime/timezone/ -H "Authorization: Bearer <token>"

{"timezone": "<Timezone>"} → Ex : {"timezone": "GMT +0000"}

curl -v https://<IP>/api/general/datetime/ntp_use/ -H "Authorization: Bearer <token>"

curl -X PUT https://<IP>/api/general/datetime/ntp_use/ -H "Authorization: Bearer <token>"

{"ntp_use": "<true/false>"} → {"ntp_use": "true"}

curl -v https://<IP>/api/general/datetime/ntp_pri_server/ -H "token:<Token String>"

curl -X PUT https://<IP>/api/general/datetime/ntp_pri_server/ -H "Authorization: Bearer <token>"

{"ntp_pri_server": "<NTP Server>"} → {"ntp_pri_server": "TIME1.google.com"}

curl -v https://<IP>/api/general/datetime/ntp_sec_server/ -H "Authorization: Bearer <token>"

curl -X PUT https://<IP>/api/general/datetime/ntp_sec_server/ -H "Authorization: Bearer <token>"

{"ntp_sec_server": "<NTP Server>"} → {"ntp_sec_server": "TIME1.google.com"}

curl -v https://<IP>/api/general/datetime/dst/ -H "Authorization: Bearer <token>"

curl -X PUT https://<IP>/api/general/datetime/dst/ -H "Authorization: Bearer <token>"

{"dst": "<none/tradition/manual>"} → {"dst": "none"}

```
curl -v https://<IP>/api/general/datetime/date_format/ -H "Authorization: Bearer <token>"
```

```
curl -X PUT https://<IP>/api/general/datetime/date_format/ -H "Authorization: Bearer <token>"
```

```
{"date_format":"<Date Format>} → {"date_format":"yyyy/mm/dd"}
```

⇒ "date_format" → "mm/dd/yyyy" 、 "yyyy/mm/dd" 、 "dd.mm.yyyy" 、 "mmm-dd-yy" 、 "dd-mmm-yy" 、 "yyyy-mm-dd"

```
curl -X PUT https://<IP>/api/general/datetime/ -H "Authorization: Bearer <token>"
```

```
Ex : {"date":"11/30/2023","time":"15:56:21"}
```

```
curl -v https://<IP>/api/general/ident/ -H "Authorization: Bearer <token>"
```

```
curl -v https://<IP>/api/general/ident/name/ -H "Authorization: Bearer <token>"
```

```
curl -X PUT https://<IP>/api/general/ident/name/ -H "Authorization: Bearer <token>"
```

```
-d '{"name":"<Name>"} → {"name":"test_name"}
```

```
curl -v https://<IP>/api/general/ident/location/ -H "Authorization: Bearer <token>"
```

```
curl -X PUT https://<IP>/api/general/ident/location/ -H "Authorization: Bearer <token>"
```

```
-d '{"location":"<Location>"} → {"location":"test_location"}
```

```
curl -v https://<IP>/api/general/ident/contact/ -H "Authorization: Bearer <token>"
```

```
curl -X PUT https://<IP>/api/general/ident/contact/ -H "Authorization: Bearer <token>"
```

```
-d '{"contact":"<Contact>"} → {"contact":"test_contact"}
```

```
curl -X PUT https://<IP>/api/general/ident/ -H "Authorization: Bearer <token>"
```

```
-d '{"name":"test_name","location":"test_location","contact":"test"}
```

5. Security

curl -v https://<IP>/api/security/ -H "Authorization: Bearer <token>"

Example Response

```
{
  "management": {
    "authtype": "localonly",
    "secret": "powerpanel.encryption.key",
    "mgrip": {
      "admin": [
        {
          "ip": "0.0.0.0",
          "access": "true"
        },
        {
          "ip": "0.0.0.0",
          "access": "false"
        }
      ],
      "viewer": [
        {
          "ip": "0.0.0.0",
          "access": "true"
        },
        {
          "ip": "0.0.0.0",
          "access": "false"
        }
      ]
    }
  },
  "local": {
    "admin": {
      "username": "admin",
      "passwd": "*****"
    },
    "viewer": {
      "access": "false",
      "username": "device",
      "passwd": "*****"
    }
  },
  "radius": {
    "num": 0,
    "server": []
  },
  "ldap": {
    "num": 0,
    "server": []
  },
  "session": {
    "timeout": 3
  }
}
```

- Show authtype of system

```
curl -v https://<IP>/api/security/management/authtype/ -H  
"Authorization: Bearer <token>"
```

- Set authtype of system

```
curl -X PUT https://<IP>/api/security/management/authtype/ -d  
'{"authtype": "localonly"}' -H "Authorization: Bearer <token>"
```

⇒ "authtype" → "localonly" 、 "radiusonly" 、 "radiuslocal" 、 "ldaponly" 、 "ldaplocal"

- Show secret of software authentication

```
curl -v https://<IP>/api/security/management/secret/ -H  
"Authorization: Bearer <token>"
```

- Set secret of software authentication

```
curl -X PUT https://<IP>/api/security/management/secret/ -d  
'{"secret": "<PPB Secret Phase>"}' -H "Authorization: Bearer <token>"
```

- Show manager IP

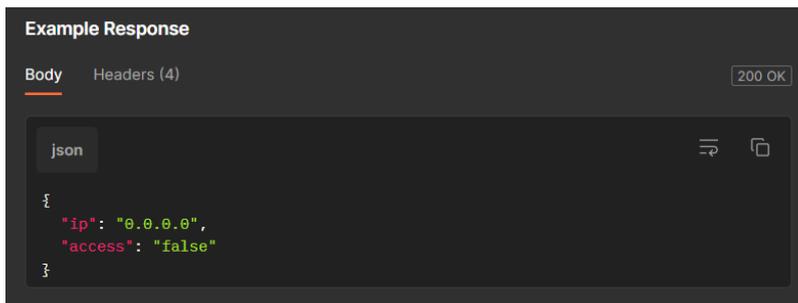
```
curl -v https://<IP>/api/security/management/mgrip/ -H  
"Authorization: Bearer <token>"
```

```
Example Response  
  
{  
  "admin": [  
    {  
      "ip": "0.0.0.0",  
      "access": "true"  
    },  
    {  
      "ip": "0.0.0.0",  
      "access": "false"  
    }  
  ],  
  "viewer": [  
    {  
      "ip": "0.0.0.0",  
      "access": "true"  
    },  
    {  
      "ip": "0.0.0.0",  
      "access": "false"  
    }  
  ]  
}
```

- Show admin secondary manager IP

```
curl -v https://<IP>/api/security/management/mgrip/admin/2/ -H
```

"Authorization: Bearer <token>"



- Set admin secondary manager IP

```
curl -X PUT https://<IP>/api/security/management/mgrip/admin/2/ -d '{"ip": "192.168.202.44", "access": "true"}' -H "Authorization: Bearer <token>"
```

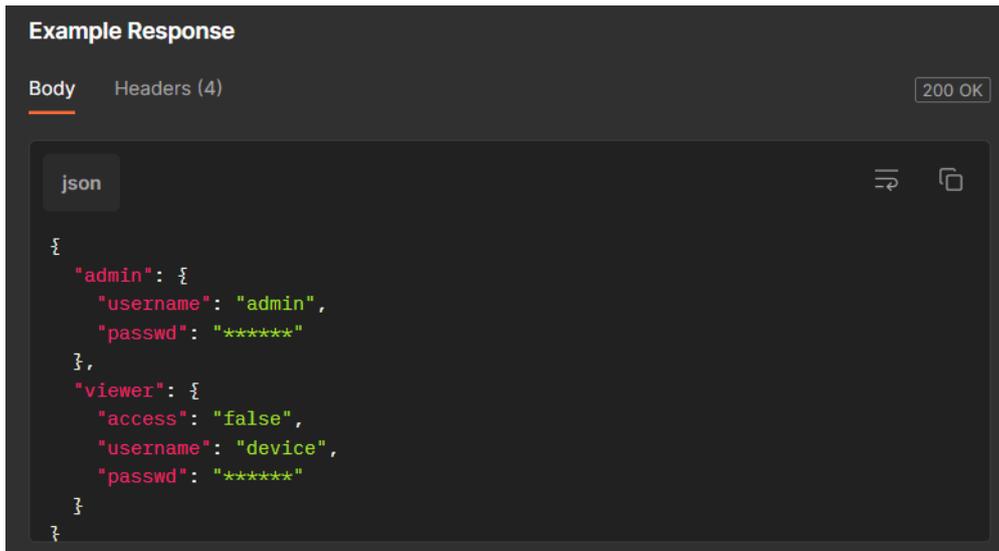
- Set manager IP by array

```
curl -X PUT https://<IP>/api/security/management/mgrip/ -d '{"mgrip": {"admin": [{"ip": "192.168.0.1", "access": "true"}, {"ip": "192.168.0.2", "access": "true"}], "viewer": [{"ip": "192.168.0.3", "access": "true"}, {"ip": "192.168.0.4", "access": "true"}]}}' -H "Authorization: Bearer <token>"
```

(Set admin primary manager IP as **192.168.0.1** , Set admin secondary manager IP as **192.168.0.2** , Set viewer primary manager IP as **192.168.0.3** , Set viewer secondary manager IP as **192.168.0.4**)

- Show Local Account

```
curl -v https://<IP>/api/security/local/ -H "Authorization: Bearer <token>"
```



- Show Admin Account

```
curl -v https://<IP>/api/security/local/admin/ -H "Authorization: Bearer <token>"
```

- Modify admin account

```
curl -X PUT https://<IP>/api/security/local/admin/ -d '{"username": "admin", "curpasswd": "admin", "passwd": "123"}' -H "Authorization: Bearer <token>"
```

- Show viewer account

```
curl -v https://<IP>/api/security/local/viewer/ -H "Authorization: Bearer <token>"
```

- Add viewer account

```
curl -X PUT https://<IP>/api/security/local/viewer/ -d '{"access": "false", "username": "device", "passwd": "cyber"}' -H "Authorization: Bearer <token>"
```

- Show radius server

```
curl -v https://<IP>/api/security/radius/ -H "Authorization: Bearer <token>"
```

Example Response

```
json
{
  "num": 1,
  "server": [
    {
      "hostname": "1.1.1.1",
      "secret": "*****",
      "port": 456,
      "authtype": "pap",
      "timeout": 20
    }
  ]
}
```

- Show the number of radius server

```
curl -v https://<IP>/api/security/radius/num/ -H "Authorization: Bearer <token>"
```

- Add radius server

```
curl -X POST https://<IP>/api/security/radius/server/ -d '{"hostname": "1.1.1.1", "secret": "cybertest111", "port": 456, "authtype": "pap", "timeout": 20}' -H "Authorization: Bearer <token>"
```

⇒ "hostname" → <STRING>

⇒ "port" → <NUMBER>

⇒ "secret" → <STRING>

⇒ "authtype" → "pap" 、 "chap"

- Modify primary radius server hostname

```
curl -X PUT https://<IP>/api/security/radius/server/1/ -d '{"hostname": "1.1.1.1", "secret": "cybertest111", "port": 456, "authtype": "pap", "timeout": 20}' -H "Authorization: Bearer <token>"
```

- Delete primary radius server

```
curl -X DELETE https://<IP>/api/security/radius/server/ -d '{"index":1}' -H "Authorization: Bearer <token>"
```

- Show ldap server

```
curl -v https://<IP>/api/security/ldap/ -H "Authorization: Bearer <token>"
```

```
Example Response

json
{
  "num": 1,
  "server": [
    {
      "hostname": "1.1.1.1",
      "ssl": "false",
      "port": 389,
      "basedn": "dc=cyber,dc=com",
      "userattr": "cn",
      "auth_n_mode": "anonymous",
      "accrdn": "",
      "accrpw": "",
      "auth_z_mode": "byattr",
      "adminattr": "description",
      "adminvalue": "cyber_admin",
      "groupbase": "",
      "groupattr": "",
      "groupvalue": "",
      "type": "generic",
      "addomain": ""
    }
  ]
}
```

- Add ldap server

```
curl -X POST https://<IP>/api/security/ldap/server/ -d '{"hostname": "1.1.1.1","ssl": "false","port": 389,"basedn": "dc=cyber,dc=com","userattr": "cn","auth_n_mode": "anonymous","auth_z_mode": "byattr","adminattr": "description","adminvalue": "cyber_admin","type": "generic"}' -H "Authorization: Bearer <token>"
```

- ⇒ "type" → "generic" 、 "ad"
- ⇒ "hostname" → <STRING>
- ⇒ "basedn" → <STRING>
- ⇒ "userattr" → <STRING>
- ⇒ "port" → <NUMBER>
- ⇒ "ssl" → "true" 、 "false"
- ⇒ "auth_n_mode" → "anonymous" 、 "user" 、 "logon"
- ⇒ "addomain" → <STRING>
- ⇒ "accrdn" → <STRING>
- ⇒ "accrpw" → <STRING>

- ⇒ "auth_z_mode" → "byattr" 、 "bygroup"
- ⇒ "adminattr" → <STRING>
- ⇒ "adminvalue" → <STRING>
- ⇒ "groupbase" → <STRING>
- ⇒ "groupattr" → <STRING>
- ⇒ "groupvalue" → <STRING>
- ⇒ "type" → "generic" 、 "ad"

- Set primary ldap server SSL as true

```
curl -X PUT https://<IP>/api/security/ldap/server/1/ -d '{"ssl":"true"}' -H "Authorization: Bearer <token>"
```

- Set primary ldap server userattr as uid

```
curl -X PUT https://<IP>/api/security/ldap/server/1/ -d '{"userattr":"uid"}' -H "Authorization: Bearer <token>"
```

- Show primary ldap server

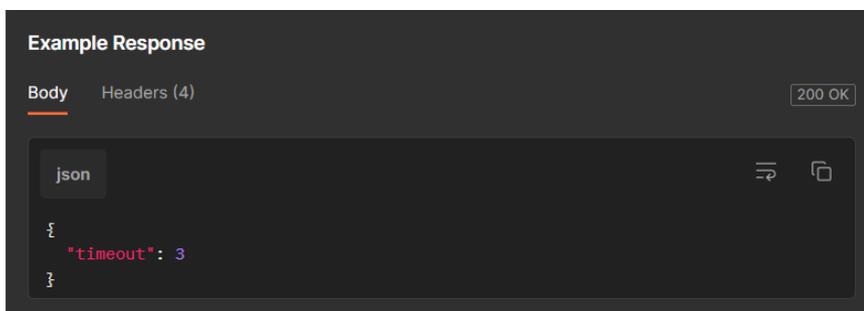
```
curl -v https://<IP>/api/security/ldap/server/1/ -H "Authorization: Bearer <token>"
```

- Delete primary ldap server

```
curl -X DELETE https://<IP>/api/security/ldap/server/ -d '{"index":1}' -H "Authorization: Bearer <token>"
```

- Show information of session

```
curl -v https://<IP>/api/security/session/ -H "Authorization: Bearer <token>"
```



- Set session timeout

```
curl -X PUT https://<IP>/api/security/session/timeout/ -d '{"timeout":5}' -H "Authorization: Bearer <token>"
```

6. Network

curl -v https://<IP>/api/network/ -H "Authorization: Bearer <token>"

```
Example Response

json

{
  "ipv4": {
    "ip": "192.168.202.56",
    "subnetmask": "255.255.255.0",
    "gateway": "192.168.202.254",
    "dns": "192.168.20.125",
    "dhcp": "true",
    "dnsfromdhcp": "true",
    "hostname": "imc05F3D3",
    "hostnamesync": "false"
  },
  "ipv6": {
    "access": "false",
    "routercontrol": "",
    "manual": "false",
    "linklocal": "",
    "manualaddr": ""
  },
  "snmpv1": {
    "access": "false",
    "user": [
      {
        "community": "public",
        "ip": "0.0.0.0",
        "accesstype": "read-only"
      },
      {
        "community": "private",
        "ip": "0.0.0.0",
        "accesstype": "read-write"
      },
      {
        "community": "public2",
        "ip": "0.0.0.0",
        "accesstype": "forbidden"
      },
      {
        "community": "public3",
        "ip": "0.0.0.0",
        "accesstype": "forbidden"
      }
    ]
  },
  "snmpv3": {
    "access": "false",
    "user": [
      {
        "name": "cyber snmpv3 user1",
        "status": "disable",
        "authprotocol": "none",
        "authpasswd": "",
        "privprotocol": "none",
        "privpasswd": "",
        "ip": "0.0.0.0"
      },
      {
        "name": "cyber snmpv3 user2",
        "status": "disable",
        "authprotocol": "none",
        "authpasswd": "",
        "privprotocol": "none",
        "privpasswd": "",
        "ip": "0.0.0.0"
      },
      {
        "name": "cyber snmpv3 user3",
        "status": "disable",
        "authprotocol": "none",
        "authpasswd": "",
        "privprotocol": "none",
        "privpasswd": "",
        "ip": "0.0.0.0"
      },
      {
        "name": "cyber snmpv3 user4",
        "status": "disable",
        "authprotocol": "none",
        "authpasswd": "",
        "privprotocol": "none",
        "privpasswd": "",
        "ip": "0.0.0.0"
      }
    ]
  }
}
```

```

"web": {
  "access": "http",
  "httpport": 80,
  "httpsport": 443,
  "alg": [
    {
      "name": "TLS_DHE_RSA_WITH_AES_256_CBC_SHA",
      "status": "true"
    },
    {
      "name": "TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA",
      "status": "true"
    },
    {
      "name": "TLS_RSA_WITH_AES_256_CBC_SHA",
      "status": "true"
    },
    {
      "name": "TLS_RSA_WITH_CAMELLIA_256_CBC_SHA",
      "status": "true"
    },
    {
      "name": "TLS_RSA_WITH_AES_128_CBC_SHA",
      "status": "true"
    },
    {
      "name": "TLS_RSA_WITH_CAMELLIA_128_CBC_SHA",
      "status": "true"
    },
    {
      "name": "TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256",
      "status": "true"
    },
    {
      "name": "TLS_DHE_RSA_WITH_AES_128_CBC_SHA",
      "status": "true"
    },
    {
      "name": "TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384",
      "status": "true"
    },
    {
      "name": "TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA",
      "status": "true"
    },
    {
      "name": "TLS_DHE_RSA_WITH_AES_256_GCM_SHA384",
      "status": "true"
    },
    {
      "name": "TLS_RSA_WITH_AES_256_CBC_SHA256",
      "status": "true"
    },
    {
      "name": "TLS_RSA_WITH_AES_128_CBC_SHA256",
      "status": "true"
    }
  ]
},
"console": {
  "access": "ssh",
  "telnetport": 23,
  "sshport": 22
},
"ftp": {
  "ftpaccess": "true",
  "ftpport": 21
},
"modbustcp": {
  "status": "offline",
  "access": "disable",
  "ip": "0.0.0.0",
  "port": 502
}
}

```

- Show IPv4 IP Address

```
curl -v https://<IP>/api/network/ipv4/ip/ -H "Authorization: Bearer <token>"
```

- Show IPv4 DHCP

```
curl -v https://<IP>/api/network/ipv4/dhcp/ -H "Authorization: Bearer <token>"
```

- Enable IPv6 access

```
curl -X PUT https://<IP>/api/network/ipv6/ -d '{"access":"true"}' -H "Authorization: Bearer <token>"
```

- Show SNMPv1 information

```
curl -v https://<IP>/api/network/snmpv1/ -H "Authorization: Bearer <token>"
```

```
Example Response

json

{
  "access": "false",
  "user": [
    {
      "community": "public",
      "ip": "0.0.0.0",
      "accesstype": "read-only"
    },
    {
      "community": "private",
      "ip": "0.0.0.0",
      "accesstype": "read-write"
    },
    {
      "community": "public2",
      "ip": "0.0.0.0",
      "accesstype": "forbidden"
    },
    {
      "community": "public3",
      "ip": "0.0.0.0",
      "accesstype": "forbidden"
    }
  ]
}
```

- Enable SNMPv1 access

```
curl -X PUT https://<IP>/api/network/snmpv1/ -d '{"access":"true"}' -H "Authorization: Bearer <token>"
```

- Set SNMPv1 secondary user IP as 192.168.0.201.

```
curl -X PUT https://<IP>/api/network/snmpv1/user/2/ -d  
'{"ip":"192.168.0.201"}' -H "Authorization: Bearer <token>"
```

⇒ "community" → <String>

⇒ "ip" → <String>

⇒ "accesstype" → "read-only" 、 "read-write" 、 "forbidden"

- Show SNMPv3 information

```
curl -v https://<IP>/api/network/snmpv3/ -H "Authorization: Bearer  
<token>"
```

Example Response

json

```
{  
  "access": "false",  
  "user": [  
    {  
      "name": "cyber snmpv3 user1",  
      "status": "disable",  
      "authprotocol": "none",  
      "authpasswd": "",  
      "privprotocol": "none",  
      "privpasswd": "",  
      "ip": "0.0.0.0"  
    },  
    {  
      "name": "cyber snmpv3 user2",  
      "status": "disable",  
      "authprotocol": "none",  
      "authpasswd": "",  
      "privprotocol": "none",  
      "privpasswd": "",  
      "ip": "0.0.0.0"  
    },  
    {  
      "name": "cyber snmpv3 user3",  
      "status": "disable",  
      "authprotocol": "none",  
      "authpasswd": "",  
      "privprotocol": "none",  
      "privpasswd": "",  
      "ip": "0.0.0.0"  
    },  
    {  
      "name": "cyber snmpv3 user4",  
      "status": "disable",  
      "authprotocol": "none",  
      "authpasswd": "",  
      "privprotocol": "none",  
      "privpasswd": "",  
      "ip": "0.0.0.0"  
    }  
  ]  
}
```

- Enable SNMPv3 access
curl **-X PUT** https://<IP>/api/network/snmpv3/ -d '{"access":"true"}' -H "Authorization: Bearer <token>"

- Show SNMPv3 forth user information
curl **-v** https://<IP>/api/network/snmpv3/user/4/ -H "Authorization: Bearer <token>"

- Set SNMPv3 forth user IP as 192.168.0.201
curl **-X PUT** https://<IP>/api/network/snmpv3/user/4/ -d '{"ip":"192.168.0.201"}' -H "Authorization: Bearer <token>"

- Set SNMPv3 secondary snmp user authentication protocol as md5 and authentication password as 1111111111111111.
curl **-X PUT** https://<IP>/api/network/snmpv3/user/2/ -d '{"authprotocol":"md5","authpasswd":"1111111111111111"}' -H "Authorization: Bearer <token>"

- Set SNMPv3 primary snmp user IP as 192.168.0.202 and set secondary snmp user IP as 192.168.0.203

```
curl -X PUT https://<IP>/api/network/snmpv3/user/ -d  
'{"user":[{"ip":"192.168.0.202"}, {"ip":"192.168.0.203"}]}' -H  
"Authorization: Bearer <token>"
```

- ⇒ "name" → <String>
- ⇒ "status" → "enable" 、 "disable"
- ⇒ "authprotocol" → "md5" 、 "sha"
- ⇒ "authpasswd" → <String>
- ⇒ "privprotocol" → "des" 、 "aes"
- ⇒ "privpasswd" → <String>
- ⇒ "ip" → <String>

- Show information of Web

```
curl -v https://<IP>/api/network/web/ -H "Authorization: Bearer  
<token>"
```

```
{  
  "access": "http",  
  "httpport": 80,  
  "httpsport": 443,  
  "alg": [  
    {  
      "name": "TLS_DHE_RSA_WITH_AES_256_CBC_SHA",  
      "status": "true"  
    },  
    {  
      "name": "TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA",  
      "status": "true"  
    },  
    {  
      "name": "TLS_RSA_WITH_AES_256_CBC_SHA",  
      "status": "true"  
    },  
    {  
      "name": "TLS_RSA_WITH_CAMELLIA_256_CBC_SHA",  
      "status": "true"  
    },  
    {  
      "name": "TLS_RSA_WITH_AES_128_CBC_SHA",  
      "status": "true"  
    },  
    {  
      "name": "TLS_RSA_WITH_CAMELLIA_128_CBC_SHA",  
      "status": "true"  
    },  
    {  
      "name": "TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256",  
      "status": "true"  
    }  
  ]  
}
```

- Set HTTP port as 5000

```
curl -X PUT https://<IP>/api/network/web/httpport/ -d  
'{"httpport":5000}' -H "Authorization: Bearer <token>"
```

- Disable TLS_DHE_DSS_WITH_AES_128_CBC_SHA algorithm of HTTPS

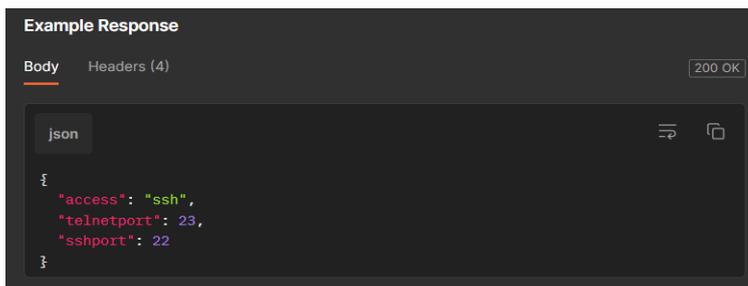
```
curl -X PUT https://<IP>/api/network/web/alg/1/ -d '{"status":"false"}'  
-H "Authorization: Bearer <token>"
```

- Upload HTTPS certification

```
curl -F upfile=@<Cert file>  
https://<IP>/api/network/web/https/upload/cert/ -H "Authorization:  
Bearer <token>"
```

- Show information of console

```
curl -v https://<IP>/api/network/console/ -H "Authorization: Bearer  
<token>"
```



- Disable Telnet and Enable SSH

```
curl -X PUT https://<IP>/api/network/console/ -d '{"access":"ssh"}' -H  
"Authorization: Bearer <token>"
```

⇒ "access" → "disable" 、 "ssh" 、 "telnet"

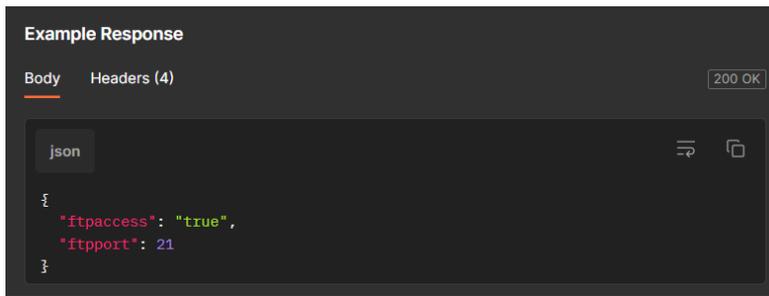
- Upload SSH hostkey

```
curl -F upfile=@<Hostkey file>  
https://<IP>/api/network/console/upload/hostkey/ -H "Authorization:  
Bearer <token>"
```

- Show information of FTP

```
curl -v https://<IP>/api/network/ftp/ -H "Authorization: Bearer
```

<token>"



- Disable FTP

```
curl -X PUT https://<IP>/api/network/ftp/ -d '{"ftppass":"false"}' -H "Authorization: Bearer <token>"
```

- Show information of Modbus TCP

```
curl -v https://<IP>/api/network/modbustcp/ -H "Authorization: Bearer <token>"
```

- Change Modbus TCP access status

```
curl -X PUT https://<IP>/api/network/modbustcp/ -d '{"access":"readonly"}' -H "Authorization: Bearer <token>"
```

⇒ "access" → "disable" 、 "readonly" 、 "readwrite"

7. Notification

curl -v https://<IP>/api/notification/ -H "Authorization: Bearer <token>"

```
{
  "event": {
    "security": [
      {
        "index": 61,
        "msg": "Login authorization failure via HTTP",
        "log": "true",
        "email": "false",
        "trap": "false",
        "syslog": "false",
        "sms": "false"
      }, {
        "index": 62,
        "msg": "Login authorization failure via Console",
        "log": "true",
        "email": "false",
        "trap": "false",
        "syslog": "false",
        "sms": "false"
      }
    ],
    "emailrcpt": {
      "num": 0,
      "rcptinfo": []
    },
    "traprcpt": {
      "num": 0,
      "rcptinfo": []
    },
    "sms": {
      "service": "clickatellold",
      "username": "Click_Name",
      "passwd": "Click_Pass",
      "apiid": "Click_api_ID",
      "api": "Click_API",
      "geturl": "",
      "posturl": "",
      "postcontent": "",
      "emailaddr": "",
      "emailsubject": "",
      "emailcontent": ""
    },
    "smsrcpt": {
      "num": 0,
      "rcptinfo": []
    }
  }
}
```

- Show security event

curl -v https://<IP>/api/notification/event/security/ -H "Authorization: Bearer <token>"

Example Response

```

json
{
  "security": [
    {
      "index": 61,
      "msg": "Login authorization failure via HTTP from [IP Address]",
      "log": "true",
      "email": "false",
      "trap": "false",
      "syslog": "false",
      "sms": "false"
    },
    {
      "index": 62,
      "msg": "The password has been changed by [IP Address]",
      "log": "true",
      "email": "false",
      "trap": "false",
      "syslog": "false",
      "sms": "false"
    },
    {
      "index": 63,
      "msg": "Login authorization failure via Console from [IP Address]",
      "log": "true",
      "email": "false",
      "trap": "false",
      "syslog": "false",
      "sms": "false"
    },
    {
      "index": 64,
      "msg": "Configuration file uploaded by [IP Address]",
      "log": "true",
      "email": "false",
      "trap": "false",
      "syslog": "false",
      "sms": "false"
    }
  ]
}

```

- Enable send "Event "Login authentication failure via HTTP" via trap
 curl -X PUT https://<IP>/api/notification/event/security/1/ -d '{"trap": "true"}' -H "Authorization: Bearer <token>"
- Enable send "Event "The password has been change" via email and syslog
 curl -X PUT https://<IP>/api/notification/event/security/ -d '{"security": [{},{ "email": "true", "syslog": "true"}],{},{}}' -H "Authorization: Bearer <token>"
- Show SMTP server
 curl -v https://<IP>/api/notification/smtp/ -H "Authorization: Bearer <token>"

Example Response

Body Headers (4) 200 OK

```

json
{
  "server": "0.0.0.0",
  "senderemail": "",
  "sendername": "",
  "auth": "false",
  "account": "",
  "passwd": "",
  "encrypt": "none",
  "port": 25
}

```

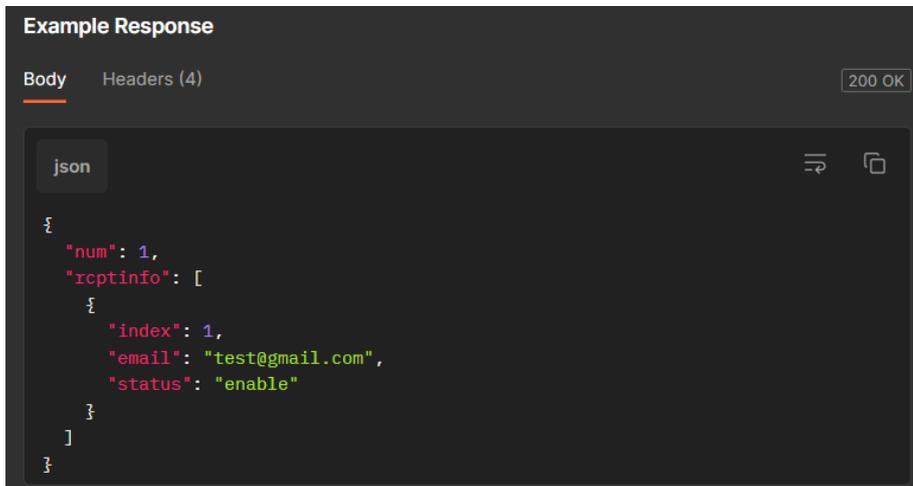
- Set SMTP server

```
curl -X PUT https://<IP>/api/notification/smtp/ -d '{"server":  
"smtp.gmail.com","senderemail": "test@gmail.com","sendername":  
"test_name","auth": "true","account": "test@gmail.com","passwd":  
"abcdefg","encrypt": "tls","port": 587}' -H "Authorization: Bearer  
<token>"
```

- ⇒ "server" → <String>
- ⇒ "senderemail" → <String>
- ⇒ "sendername" → <String>
- ⇒ "auth" → "true" 、 "false"
- ⇒ "account" → <String>
- ⇒ "passwd" → <String>
- ⇒ "encrypt" → "none" 、 "tls" 、 "ssl"
- ⇒ "port" → <Number>

- Show e-mail recipients

```
curl -v https://<IP>/api/notification/emailrcpt/ -H "Authorization:  
Bearer <token>"
```



- Add e-mail recipient

```
curl -X POST https://<IP>/api/notification/emailrcpt/rcptinfo/ -d  
'{"status":"enable","email":"test@gmail.com"}' -H "Authorization:  
Bearer <token>"
```

- Delete first e-mail recipient

```
curl -X DELETE https://<IP>/api/notification/emailrcpt/rcptinfo/ -d  
'{"index":1}' -H "Authorization: Bearer <token>"
```

- Disable first e-mail recipient

```
curl -X PUT https://<IP>/api/notification/emailrcpt/rcptinfo/1/ -d '{"status":"disable"}' -H "Authorization: Bearer <token>"
```

⇒ "status" → "enable" 、 "disable"

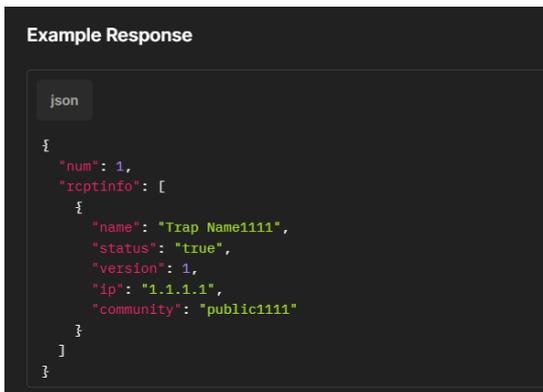
⇒ "email" → <String>

- Send test message to first email recipient

```
curl -X POST https://<IP>/api/notification/emailrcpt/sendtest/ -d '{"index":1}' -H "Authorization: Bearer <token>"
```

- Show trap recipients

```
curl -v https://<IP>/api/notification/traprcpt/ -H "Authorization: Bearer <token>"
```



The screenshot shows a terminal window titled "Example Response" with a dark background. The output is a JSON object representing a trap recipient. The JSON is as follows:

```
json
{
  "num": 1,
  "rcptinfo": [
    {
      "name": "Trap Name1111",
      "status": "true",
      "version": 1,
      "ip": "1.1.1.1",
      "community": "public1111"
    }
  ]
}
```

- Add SNMPv1 trap receiver

```
curl -X POST https://<IP>/api/notification/traprcpt/rcptinfo/ -d '{"name":"testname","ip":"192.168.0.202","status":"true","version":1,"community":"testcomm"}' -H "Authorization: Bearer <token>"
```

- Add SNMPv3 trap receiver

```
curl -X POST https://<IP>/api/notification/traprcpt/rcptinfo/ -d '{"name":"testname","ip":"192.168.0.202","status":"true","version":3,"user_idx":1}' -H "Authorization: Bearer <token>"
```

- Delete first trap receiver

```
curl -X DELETE https://<IP>/api/notification/traprcpt/rcptinfo/ -d '{"index":1}' -H "Authorization: Bearer <token>"
```

- Set first trap receiver name as "testname"

```
curl -X PUT https://<IP>/api/notification/traprcpt/rcptinfo/1/ -d  
'{"name":"testname"}' -H "Authorization: Bearer <token>"
```

⇒ "name" → <String>

⇒ "status" → "true" 、 "false"

⇒ "version" → 1 、 3

⇒ "ip" → <String>

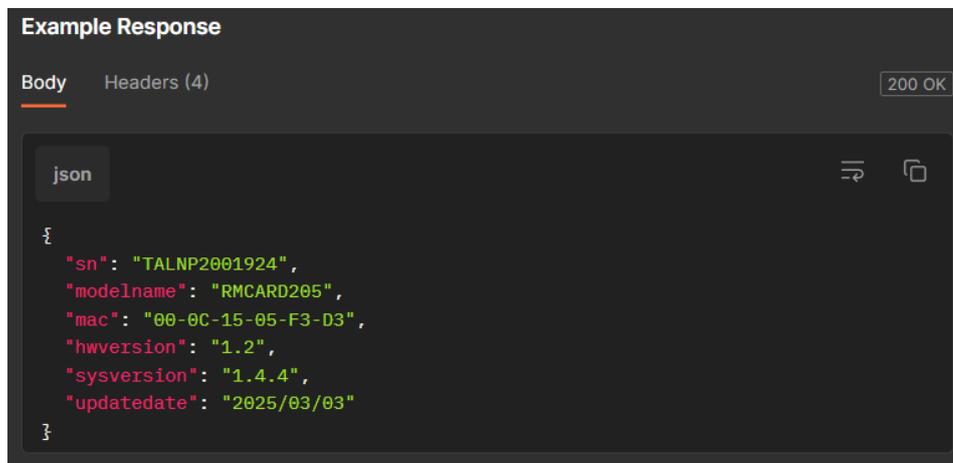
⇒ "community" → <String>

- Send test message to first trap receiver

```
curl -X POST https://<IP>/api/notification/traprcpt/sendtest/ -d  
'{"index":1}' -H "Authorization: Bearer <token>"
```

8. System

```
curl -v https://<IP>/api/system/ -H "Authorization: Bearer <token>"
```



- Show model name

```
curl -v https://<IP>/api/system/modelname/ -H "Authorization: Bearer <token>"
```

- Reboot system

```
curl -X POST https://<IP>/api/system/reboot/ -d '{"reboot":"true"}' -H "Authorization: Bearer <token>"
```

- Reset system

```
curl -X POST https://<IP>/api/system/reset/ -d '{"reset":"true"}' -H "Authorization: Bearer <token>"
```

- Reset system (TCP/IP Settings Reserved)

```
curl -X POST https://<IP>/api/system/resetnotcpip/ -d '{"resetnotcpip":"true"}' -H "Authorization: Bearer <token>"
```

- Upload restore configuration file

```
curl -F upfile=@<Restore file> https://<IP>/api/system/restore/upload/ -H "Authorization: Bearer <token>"
```

- Download save configuration file

```
curl https://<IP>/api/system/restore/download/ --output <Download File Name> -H "Authorization: Bearer <token>"
```

- Download diagnostic information file
curl https://<IP>/api/system/diagnoinfo/download/ --output <Download File Name> -H "Authorization: Bearer <token>"

- Firmware update
curl -F upfile=@<Data file> -F upfile=@<FW file>
https://<IP>/api/system/firmware/upload/ -H "Authorization: Bearer <token>"

9. UPS Status

curl -v https://<IP>/api/upsstatus/ -H "Authorization: Bearer <token>"

```
{
  "input": {
    "status": "Normal",
    "voltage": 114.7,
    "frequency": 59.9
  },
  "output": {
    "status": "Normal",
    "voltage": 110.2,
    "frequency": 59.9,
    "load": 0,
    "current": 0.0,
    "ncl": On
  },
  "battery": {
    "status": "Fully Charged",
    "capacity": 100,
    "runtime": 6300,
    "voltage": 41.0
  },
  "system": {
    "status": "Normal",
    "statusidx": 0,
    "tempc": 25,
    "tempcf": 77
  }
}
```

(1) input

- ⇒ "status" → "Over Voltage" 、 " Under Voltage" 、 " Frequency Failure" 、 " Blackout" 、 " Normal"
- ⇒ "voltage" → <Number> (V)
- ⇒ "frequency" → <Number> (Hz)

(2) output

- ⇒ "status" → "Normal" 、 " Boost" 、 " Buck" 、 " Over Load" 、 " Bypass" 、 "

Manual Bypass" 、 " Bypass Overload" 、 " ECO Mode" 、 " No Output"

- ⇒ "voltage" → <Number> (V)
- ⇒ "frequency" → <Number> (Hz)
- ⇒ "load" -> <Number> (%)
- ⇒ "current" → <Number> (A)
- ⇒ "ncl" → "On" 、 "Off" 、 "None"
- ⇒ "ncl2" → "On" 、 "Off" 、 "None"

(3) battery

- ⇒ "status" → "Normal" 、 " Discharging" 、 " Charging" 、 " Fully Charged" 、 " Not Present" 、 " Battery Testing" 、 " Battery Critically Low"
- ⇒ "chargemode" → "sbm" 、 "normal"
- ⇒ "chargestate" → "discharge" 、 " charge" 、 " float" 、 " rest"
- ⇒ "capacity" → <Number> (%)
- ⇒ "runtime" → <Number> (s)
- ⇒ "voltage" → <Number> (V)
- ⇒ "temperature" → <Number> (C)

(4) system

- ⇒ "status" → "Normal" 、 " hwfailure" 、 " overheat"
- ⇒ "errcode" → <String> (HW Fault Error Code)
- ⇒ "tempc" → <Number> (C)
- ⇒ "tmepf" → <Number> (F)

10. UPS Information

curl -v https://<IP>/api/upsinfo/ -H "Authorization: Bearer <token>"

```
{
  "model": "OL1500RT JP",
  "voltrating": "110",
  "workfreq": "40~70",
  "pwrrating": "1500",
  "currrating": "13",
  "loadpwr": "1080",
  "voltrating": 36,
  "fwversion": "Sv3AI2",
  "usbfwversion": "0.1C",
  "lcdfwversion": "",
  "battrdate": "Feb-09-28 ",
  "nclbanknum": 1,
  "exbattpacknum": 0
}
```

- ⇒ "model" → <String>
- ⇒ "sn" → <String>
- ⇒ "voltrating" → <Number> (V)
- ⇒ "workfreq" → <Number> (Hz)
- ⇒ "pwrrating" → <Number> (V)
- ⇒ "currrating" → <Number> (A)
- ⇒ "loadpwr" → <Number> (Watt)
- ⇒ "battvoltrating" → <Number> (V)
- ⇒ "fwversion" → <String>
- ⇒ "usbversion" → <String>
- ⇒ "lcdversion" → <String>
- ⇒ "battrdate" → <String>
- ⇒ "nclbanknum" → <Number>
- ⇒ "exbattpacknum" → <Number>

11. UPS Configuration

curl -v https://<IP>/api/upsconfig/ -H "Authorization: Bearer <token>"

```
{
  "suppliedpwr": {
    "suppliedvolt": 110,
    "voltlist": [100, 110, 115, 120, 125]
  },
  "pwrfailcondi": {
    "highinvthre": 150,
    "hvthrelist": [150],
    "lowinvthre": 80,
    "lvthrelist": [80],
    "fregtol": 7,
    "fregtollist": [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
  },
  "battery": {
    "lbattthre": 20,
    "lbattthrelist": [10, 20, 30],
    "periodtest": 0
  },
  "system": {
    "coldstart": "true",
    "alarm": "false",
    "dryrelaycondi": "linefail",
    "screensaver": 0,
    "screensaverlist": [0, 60, 300],
    "wfaultdetect": "false",
    "overdischargep": 0,
    "overdischlist": [0, 20, 40, 60],
    "sleepclientsd": "false"
  }
}
```

- Show supplied power of UPS configuration

curl -v https://<IP>/api/upsconfig/suppliedpwr/ -H "Authorization: Bearer <token>"

```
{
  "suppliedvolt": 110,
  "voltlist": [100, 110, 115, 120, 125]
}
```

- Set supplied power as 120V

curl -X PUT https://<IP>/api/upsconfig/suppliedpwr/ -d '{"suppliedvolt":120}' -H "Authorization: Bearer <token>"

- Show utility power failure condition of UPS configuration

curl -v https://<IP>/api/upsconfig/pwrfailcondi/ -H "Authorization: Bearer <token>"

```

{
  "highinvthre": 150,
  "hvthrelist": [150],
  "lowinvthre": 80,
  "lvthrelist": [80],
  "freqtol": 7,
  "freqtollist": [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
}

```

- Set frequency tolerance as 3 Hz

```

curl -X PUT https://<IP>/api/upsconfig/pwrfailcondi/ -d '{"freqtol":3}' -
H "Authorization: Bearer <token>"

```

- ⇒ "sensitivity" → "low" 、 " medium" 、 " high"
- ⇒ "highinvthre" → <Number> (V)
- ⇒ "highoutvthre" → <Number> (V)
- ⇒ "lowinvthre" → <Number> (V)
- ⇒ "lowoutvthre" → <Number> (V)
- ⇒ "freqtol" → <Number> (Hz)

- Show operation of UPS configuration

```

curl -v https://<IP>/api/upsconfig/operation/ -H "Authorization: Bearer
<token>"

```

```

{
  "mode": "normal",
  "exdays": [
    {
      "day": "sun",
      "status": "false"
    }, {
      "day": "mon",
      "status": "false"
    }, {
      "day": "tue",
      "status": "false"
    }, {
      "day": "wen",
      "status": "false"
    }, {
      "day": "thu",
      "status": "false"
    }, {
      "day": "fri",
      "status": "false"
    }, {
      "day": "sat",
      "status": "false"
    }
  ],
  "exhourh": 9,
  "exhourl": 18
}

```

- Set exclusive says to enable Sunday and Monday

```
curl -X PUT https://<IP>/api/upsconfig/operation/exdays/ -d '{"exdays":{"status":"true"},"status":"true"}' -H "Authorization: Bearer <token>"
```

- ⇒ "mode" → "normal" 、 "eco" 、 "eco15" 、 "eco10" 、 "generator" 、 "bypass"
- ⇒ "exhourh" → <Number> (0 ~ 23)
- ⇒ "exhourl" → <Number> (0 ~ 23)

- Show bypass of UPS configuration

```
curl -v https://<IP>/api/upsconfig/bypass/ -H "Authorization: Bearer <token>"
```

```
{
  "bypasscondi": "chkfreqvolt",
  "vupbound": 10,
  "vupboundlist": [10, 15],
  "vlowbound": 10,
  "vlowboundlist": [10, 15, 20]
}
```

- Set bypass condition as "Check Volt Only"

```
curl -X PUT https://<IP>/api/upsconfig/bypass/ -d '{"bypasscondi":"chkvoltonly"}' -H "Authorization: Bearer <token>"
```

- ⇒ "bypasscondi" → "nobypass" 、 "chkfreqvolt" 、 "chkvoltonly"
- ⇒ "vupbound" → <Number>
- ⇒ "vlowbound" → <Number>

- Show power restore of UPS configuration

```
curl -v https://<IP>/api/upsconfig/pwrrestore/ -H "Authorization: Bearer <token>"
```

```
{
  "autorestore": "true",
  "recharged": 0,
  "rechargedlist": [0, 1, 2, 3, 5, 10, 20, 30, 60],
  "rechargecap": 0,
  "rechargecaplist": [0, 15, 30, 45, 60, 75, 90],
  "returndelay": 0
}
```

- Set automatic restore as disable

```
curl -X PUT https://<IP>/api/upsconfig/pwrrestore/ -d '{"autorestore":"false"}' -H "Authorization: Bearer <token>"
```

- ⇒ "autorestore" → "true" 、 " false"
- ⇒ "recharged" → <Number>
- ⇒ "rechargecap" → <Number>
- ⇒ "returndelay" → <Number> (0 ~ 600)
- ⇒ "lsdelay" → <Number> (0 ~ 600)

- Show battery of UPS configuration

```
curl -v https://<IP>/api/upsconfig/battery/ -H "Authorization: Bearer <token>"
```

```
{
  "lbattthre": 20,
  "lbattthrelist": [10, 20, 30],
  "periodtest": 0
}
```

- Set low battery threshold as 30%

```
curl -X PUT https://<IP>/api/upsconfig/battery/ -d '{"lbattthre":30}' -H "Authorization: Bearer <token>"
```

- Set periodical battery test as 2 weeks

```
curl -X PUT https://<IP>/api/upsconfig/battery/ -d '{"periodtest":20160}' -H "Authorization: Bearer <token>"
```

- ⇒ "lbattthre" → <Number>
- ⇒ "lbruntimethre" → <Number>
- ⇒ "exmod" → "auto" 、 " manual"
- ⇒ "exbattnum" → <Number>
- ⇒ "packtype" → "standard" 、 " customized"
- ⇒ "startuptest" → "true" 、 "false"
- ⇒ "periodtest" → <Number> (min)
- ⇒ "chargemode" → "normal" 、 " sbm"
- ⇒ "chargecheck" → "true" 、 "false"

- Show system of UPS configuration
`curl -v https://<IP>/api/upsconfig/system/ -H "Authorization: Bearer <token>"`

```
{
  "coldstart": "true",
  "alarm": "false",
  "dryrelaycondi": "linefail",
  "screensaver": 0,
  "screensaverlist": [0, 60, 300],
  "overdischargep": 0,
  "overdischlist": [0, 20, 40, 60],
  "sleepclientsd": "false"
}
```

- Disable cold start
`curl -X PUT https://<IP>/api/upsconfig/system/ -d '{"coldstart":"false"}' -H "Authorization: Bearer <token>"`

- ⇒ "coldstart" → "true" 、 "false"
- ⇒ "alarm" → "true" 、 "false"
- ⇒ "dryrelaycondi" → "linefail" 、 " battlow" 、 " alarm" 、 " bypass" 、 " upsfail"
- ⇒ "screensaver" → <Number>
- ⇒ "wfaultdetect" → "true" 、 "false"
- ⇒ "overdischargep" → <Number>
- ⇒ "sleepclientsd" → "true" 、 "false"

12. UPS Master Switch

```
curl -v https://<IP>/api/upsswitch/ -H "Authorization: Bearer <token>"
```

```
{
  "sddelaylist": [0, 10, 20, 30, 60, 120, 180, 300, 600],
  "rebootdlist": [10, 20, 30, 60, 120, 180, 300, 600],
  "sleepdlist": [0, 10, 20, 30, 60, 120, 180, 300, 600]
}
```

- Turn on UPS

```
curl -X POST https://<IP>/api/upsswitch/ -d '{"turnon':"true"}' -H "Authorization: Bearer <token>"
```

- Turn off UPS (Off Delay : 10sec, Sync Remote : On)

```
curl -X POST https://<IP>/api/upsswitch/ -d '{"sddelay":10,"syncppb":"true","turnoff":"true"}' -H "Authorization: Bearer <token>"
```

- Reboot UPS (Off Delay : 10sec, Reboot Duration : 10sec)

```
curl -X POST https://<IP>/api/upsswitch/ -d '{"rebootd":10,"sddelay":10,"reboot":"true"}' -H "Authorization: Bearer <token>"
```

13. UPS Bank

curl -v https://<IP>/api/upsbank/ -H "Authorization: Bearer <token>"

```
{
  "banknum": 2,
  "bank": [{
    "type": "cl",
    "switchable": "false",
    "status": "on",
    "outletnum": 4,
    "outlet": [{
      "idx": 5,
      "name": "Outlet5"
    }, {
      "idx": 4,
      "name": "Outlet4"
    }, {
      "idx": 3,
      "name": "Outlet3"
    }, {
      "idx": 2,
      "name": "Outlet2"
    }, {
      "idx": 1,
      "name": "Outlet1"
    }
  ], {
    "status": "on",
    "outletnum": 4,
    "outlet": [
      {
        "idx": 1,
        "name": "Outlet1"
      },
      {
        "idx": 2,
        "name": "Outlet2"
      },
      {
        "idx": 3,
        "name": "Outlet3"
      },
      {
        "idx": 4,
        "name": "Outlet4"
      }
    ]
  }
  ]
}
```

- Turn off NCL Bank (Bank 2)

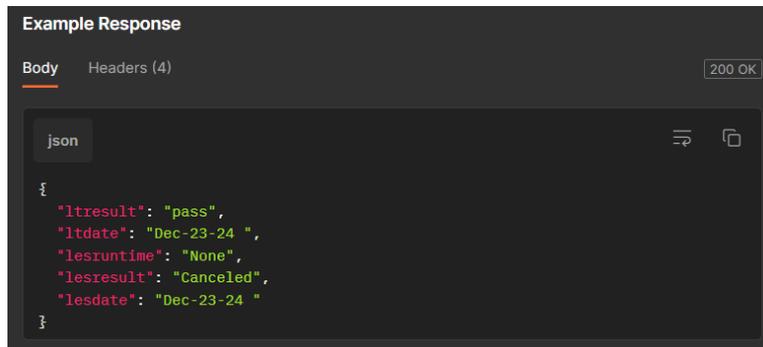
curl -X PUT https://<IP>/api/upsbank/bank/2/ -d '{"status":"off"}' -H "Authorization: Bearer <token>"

- Set first outlet name of bank 1

curl -X PUT https://<IP>/api/upsbank/bank/1/outlet/1/ -d '{"name":"test_outlet_name"}' -H "Authorization: Bearer <token>"

14. UPS Diagnostics

```
curl -v https://<IP>/api/upsdiagno/ -H "Authorization: Bearer <token>"
```



- Execute the battery test

```
curl -X PUT https://<IP>/api/upsdiagno/ -d '{"selftest":"true"}' -H "Authorization: Bearer <token>"
```

- Execute runtime estimation

```
curl -X PUT https://<IP>/api/upsdiagno/ -d '{"esstart":"true"}' -H "Authorization: Bearer <token>"
```

- Abort runtime estimation

```
curl -X PUT https://<IP>/api/upsdiagno/ -d '{"esabort":"true"}' -H "Authorization: Bearer <token>"
```

15. UPS Schedule

```
curl -v https://<IP>/api/upssche/ -H "Authorization: Bearer <token>"
```

```
{
  "schemum": 1,
  "schedule": [
    {
      "name": "Schedule Name",
      "status": "enable",
      "freq": "once",
      "bank": 255
    }
  ]
}
```

- Show first schedule

```
curl -v https://<IP>/api/upssche/schedule/1/ -H "Authorization: Bearer <token>"
```

- ⇒ "name" → <String>
- ⇒ "status" → "enable" 、 "disable"
- ⇒ "freq" → "once" 、 "daily" 、 "weekly"
- ⇒ "bank" → <Number> (255->All, 1, 2, 3)
- ⇒ "sdyear" → <Number> (>= system year)
- ⇒ "sdmonth" → <Number> (1 ~ 12)
- ⇒ "sddate" → <Number> (1 ~ 31)
- ⇒ "sdweekday" → "sun" 、 "mon" 、 "tue" 、 "wen" 、 "thu" 、 "fri" 、 "sat"
- ⇒ "sdhour" → <Number> (0 ~ 23)
- ⇒ "sdmin" → <Number> (0 ~ 59)
- ⇒ "restoretype" → "never" 、 "instant" 、 "time"
- ⇒ "reyear" → <Number>
- ⇒ "remonth" → <Number> (1 ~ 12)
- ⇒ "redate" → <Number> (1 ~ 31)
- ⇒ "rehour" → <Number> (0 ~ 23)
- ⇒ "remin" → <Number> (0 ~ 59)
- ⇒ "restored" → "same" 、 "next"
- ⇒ "sdremote" → "true" 、 "false"

16. UPS Wake on lan

```
curl -v https://<IP>/api/upswol/ -H "Authorization: Bearer <token>"
```

```
{
  "syncppb":      "true",
  "upsturnon":    "true",
  "pwrrestore":   "true",
  "remotelist":   [],
  "manuallist":   []
}
```

Add WoL Manual List

```
curl -X POST https://<IP>/api/upswol/ -d '{"status": "enable", "ip": "192.168.20.100"}' -H "Authorization: Bearer <token>"
```

17. UPS Event Log

`curl -v https://<IP>/api/upsevent/ -H "Authorization: Bearer <token>"`

```
{
  "total_num": 314,
  "start": 0,
  "event": [
    {
      "date": "2024/01/03",
      "time": "19:06:25",
      "msg": "Communication to the UPS has been established"
    }, {
      "date": "2024/01/03",
      "time": "19:06:15",
      "msg": "Admin user login from 172.17.2.110. (api user
"cyber\" )"
    }, {
      "date": "2024/01/03",
      "time": "18:05:44",
      "msg": "Configuration changed by 172.17.2.107."
    }, {
      "date": "2024/01/03",
      "time": "18:04:03",
      "msg": "Admin user login from 172.17.2.110. (api user
"cyber\" )"
    }
  ]
}
```

(Show the top 10 events)

- Show next 10 events

`curl -X PUT https://<IP>/api/upsevent/event/ -d '{"nextpage":"true"}' -H "Authorization: Bearer <token>"`

- Review events

`curl -X PUT https://<IP>/api/upsevent/event/ -d '{"review":"true"}' -H "Authorization: Bearer <token>"`

- Clear all events

`curl -X PUT https://<IP>/api/upsevent/event/ -d '{"reset":"true"}' -H "Authorization: Bearer <token>"`

18. UPS Record Data

curl -v https://<IP>/api/upsrec/ -H "Authorization: Bearer <token>"

```
{
  "total_num": 1799,
  "start": 0,
  "interval": 2,
  "intervallist": [1, 2, 5, 10, 20, 30, 60, 120, 240, 480, 720, 1440],
  "rec": [{
    "date": "2024/01/03",
    "time": "19:22:30",
    "invmin": 117.6,
    "invmax": 118.2,
    "inf": 60.0,
    "infoutv": 118.1,
    "infoutvoutf": 60.0,
    "load": 0,
    "capacity": 100,
    "runtime": 288
  }, {
    "date": "2024/01/03",
    "time": "18:56:44",
    "invmin": 118.0,
    "invmax": 118.0,
    "inf": 60.0,
    "infoutv": 118.0,
    "infoutvoutf": 60.0,
    "load": 0,
    "capacity": 100,
    "runtime": 280
  }
]}
```

(Show the top 10 records)

- Show next 10 records

curl -X PUT https://<IP>/api/upsrec/rec/ -d '{"nextpage":"true"}' -H "Authorization: Bearer <token>"

- Review records

curl -X PUT https://<IP>/api/upsrec/rec/ -d '{"review":"true"}' -H "Authorization: Bearer <token>"

- Clear all records

curl -X PUT https://<IP>/api/upsrec/rec/ -d '{"reset":"true"}' -H "Authorization: Bearer <token>"

19. Syslog

```
curl -v https://<IP>/api/syslog/ -H "Authorization: Bearer <token>"
```



```
Example Response
json
{
  "access": "true",
  "facility": 21,
  "servernum": 4,
  "server": [
    {
      "ip": "1.1.1.1",
      "port": 515
    },
    {
      "ip": "2.2.2.2",
      "port": 516
    },
    {
      "ip": "3.3.3.3",
      "port": 517
    },
    {
      "ip": "4.4.4.4",
      "port": 518
    }
  ]
}
```

- Add Syslog Server

```
curl -X POST https://<IP>/api/syslog/server/ -d '{"ip": "1.1.1.1", "port": 516}' -H "Authorization: Bearer <token>"
```

- Modify Syslog Server

```
curl -X POST https://<IP>/api/syslog/server/ -d '{"access": "true", "facility": 21, "server": [{"ip": "1.1.1.1", "port": 515}, {"ip": "2.2.2.2", "port": 516}, {"ip": "3.3.3.3", "port": 517}, {"ip": "4.4.4.4", "port": 518}]}' -H "Authorization: Bearer <token>"
```

⇒ "access" → "true", "false"

⇒ "facility" → <Number> (0 ~ 23)

⇒ "ip" → <String>

⇒ "port" → <Number>

- Delete Syslog Server
 curl -X DELETE https://<IP>/api/syslog/server/ -d '{"index": 1}' -H "Authorization: Bearer <token>"

20. Accessory

- curl -v https://<IP>/api/accessory/ -H "Authorization: Bearer <token>"

```
{
  "env": {
    "unit": "celcius",
    "num": 4,
    "device": [
      {
        "status": {
          "name": "testname",
          "location": "Server Room",
          "temp": 22.65,
          "humid": 69.53
        },
        "config": {
          "name": "testname",
          "location": "Server Room",
          "temphthres": -17,
          "templthres": -17,
          "humhthres": 0,
          "humlthres": 0
        }
      },
      .
      .
      .
      .
      .
    ],
    {
      "status": {
        "name": "EnvSensor",
        "location": "Server Room",
        "temp": 22.42,
        "humid": 69.20
      },
      "config": {
        "name": "EnvSensor",
        "location": "Server Room",
        "temphthres": 32,
        "templthres": 15,
        "humhthres": 80,
        "humlthres": 20
      }
    }
  ]
}
```

- Show information of first accessory environment sensor
 curl -v https://<IP>/api/accessory/env/device/1/ -H "Authorization: Bearer <token>"
- Show temperature unit
 curl -v https://<IP>/api/accessory/env/unit/ -H "Authorization: Bearer <token>"

- Set first accessory environment sensor name as "testname"
 curl -X PUT https://<IP>/api/accessory/env/device/1/config/ -d
 '{"name":"testname"}' -H "Authorization: Bearer <token>"

- Set temperature high threshold
 curl -X PUT https://<IP>/api/accessory/env/device/1/config/ -d
 '{"temphthres":14}' -H "Authorization: Bearer <token>"

- Set accessory basic
 curl -X PUT https://<IP>/api/accessory/env/device/2/config/ -d
 '{"temphthres":31,"templthres":14,"temphyster":4,"tempchange":9,"
 humhthres":80,"humlthres":15,"humhyster":4,"humchange":10}' -H
 "Authorization: Bearer <token>"

- Set accessory contact
 curl -X PUT https://<IP>/api/accessory/env/device/2/config/ -d
 '{"contact4name":"conname4","contact4state":"close"}' -H
 "Authorization: Bearer <token>"

- ⇒ "unit" → "celcius" 、 "fahrenheit"
- ⇒ "name" → <String>
- ⇒ "location" → <String>
- ⇒ "temphthres" → <Number>
- ⇒ "templthres" → <Number>
- ⇒ "temphyster" → <Number>
- ⇒ "tempchange" → <Number>
- ⇒ "humhthres" → <Number>
- ⇒ "humlthres" → <Number>
- ⇒ "humhyster" → <Number>
- ⇒ "humchange" → <Number>
- ⇒ "contact1name" → <String>
- ⇒ "contact1state" → "open" 、 "close"
- ⇒ "contact2name" → <String>
- ⇒ "contact2state" → "open" 、 "close"
- ⇒ "contact3name" → <String>
- ⇒ "contact3state" → "open" 、 "close"
- ⇒ "contact4name" → <String>
- ⇒ "contact4state" → "open" 、 "close"



CyberPower

CyberPower Systems, Inc.

www.cyberpowersystems.com

For USA and Canada:

4241 12th Ave East, Suite 400

Shakopee, MN 55379

Toll-free: (877) 297-6937

For all other regions:

Please visit our website for local contact information.



K01-E000133-01