



Remote Management Card

RMCARD400

REST API Manual

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

1. Login

```
curl -X POST http://<IP>/api/login/ -d  
{“username”:“<Account>”,“passwd”:“<Password>”}
```

```
{  
  "loginresult": "success",  
  "token": "19130FA93D7637CB816A04C9EF4F223B40D99879E89FBC376D50B7F5520DB8CB"  
}
```

Example for POSTMAN

The screenshot shows the Postman interface for a POST request. The URL is `http://<ip>/api/login/`. The request body is a JSON object: `{“username”:“cyber”,“passwd”:“123”}`. The response is a JSON object: `{“loginresult”:“success”,“token”:“2EB07713B68BD095D93CAD20460E320462AA786AEE015A8142827C0699DBB7D9”}`. The response status is 200 OK, 1084 ms, 303 B.

2. Logout

```
curl -X PUT http://<IP>/api/logout/ -d {“logout”:“true”} -H  
“token:<Token>”
```

3. General

curl -v http://<IP>/api/general/ -H "token:<Token>"

```
cyber@ubuntu:~$ curl -v http://172.17.2.209/api/general/ -H "token:CE23DB717E25EFD646D9848FCC4BDDC5C073D0A844D55739E804B6A802B0168A"
```

```
{
  "datetime": {
    "date": "11/30/2023",
    "time": "10:57:47",
    "timezone": "London",
    "ntp_use": "false",
    "ntp_pri_server": "0.0.0.0",
    "ntp_sec_server": "0.0.0.0",
    "dst": "false",
    "date_format": "mm/dd/yyyy"
  },
  "ident": {
    "name": "RMCARD400",
    "location": "Server Room",
    "contact": "Administrator"
  }
}
```

- show ident

curl -v http://<IP>/api/general/ident/ -H "token:<Token>"

```
{
  "name": "RMCARD400",
  "location": "Server Room",
  "contact": "Administrator"
}
```

- set identification name

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

The rest follows the same pattern

curl -v http://<IP>/api/general/datetime/ -H "token:<Token>"

curl -v http://<IP>/api/general/datetime/ date/ -H "token:<Token>"

curl -X POST http://<IP>/api/general/datetime/ date/ -H "token:<Token>"
'{"date":"mm/dd/yyyy"}' → Ex : {"date":"11/30/2023"}

curl -v http://<IP>/api/general/datetime/ time/ -H "token:<Token>"

curl -X PUT http://<IP>/api/general/datetime/ time/ -H "token:<Token>"
'{"time":"hh:mm:ss"}' → Ex : {"time":"14:32:45"}

curl -v http://<IP>/api/general/datetime/ timezone/ -H "token:<Token>"

curl -X PUT http://<IP>/api/general/datetime/ timezone/ -H "token:<Token>"
'{"timezone":"<City>"}' → Ex : {"timezone": "London"}

curl -v http://<IP>/api/general/datetime/ ntp_use/ -H "token:<Token>"

```

curl -X PUT http://<IP>/api/general/datetime/ ntp_use/ -H "token:<Token>"
  {"ntp_use":"<true/false>} → {"ntp_use":"true"}
curl -v http://<IP>/api/general/datetime/ ntp_pri_server/ -H "token:<Token String>"
curl -X PUT http://<IP>/api/general/datetime/ ntp_pri_server/ -H
"token:<Token>"
  {"ntp_pri_server":"<NTP Server>} → {"ntp_pri_server":"TIME1.google.com"}
curl -v http://<IP>/api/general/datetime/ ntp_sec_server/ -H "token:<Token>"
curl -X PUT http://<IP>/api/general/datetime/ ntp_sec_server/ -H
"token:<Token>"
  {"ntp_sec_server":"<NTP Server>} → {"ntp_sec_server":"TIME1.google.com"}
curl -v http://<IP>/api/general/datetime/ dst/ -H "token:<Token>"
curl -X PUT http://<IP>/api/general/datetime/ dst/ -H "token:<Token>"
  {"dst":"<true/false>} → {"dst":"true"}
curl -v http://<IP>/api/general/datetime/ date_format/ -H "token:<Token>"
curl -X PUT http://<IP>/api/general/datetime/ date_format/ -H "token:<Token>"
  {"date_format":"<Date Format>} → {"date_format":"yyyy/mm/dd"}

```

```

curl -X PUT http://<IP>/api/general/datetime/ -H "token:<Token>"
  Ex : {"date":"11/30/2023","time":"15:56:21"}

```

```

curl -v http://<IP>/api/general/ident/ -H "token:<Token>"
curl -v http://<IP>/api/general/ident/ name/ -H "token:<Token>"
curl -X PUT http://<IP>/api/general/ident/ name/ -H "token:<Token>"
  -d {"name":"<Name>} → {"name":"test_name"}
curl -v http://<IP>/api/general/ident/ location/ -H "token:<Token>"
curl -X PUT http://<IP>/api/general/ident/ location/ -H "token:<Token>"
  -d {"location":"<Location>} → {"location":"test_location"}
curl -v http://<IP>/api/general/ident/ contact/ -H "token:<Token>"
curl -X PUT http://<IP>/api/general/ident/ contact/ -H "token:<Token>"
  -d {"contact":"<Contact>} → {"contact":"test_contact"}

```

```

curl -X PUT http://<IP>/api/general/ident/ -H "token:<Token>"
  -d {"name":"test_name","location":"test_location","contact":"test"}

```

4. Security

curl -v http://<IP>/api/security/ -H "token:<Token>"

```
{
  "management": {
    "authtype": "localonly",
    "secret": "powerpanel.encryption.key",
    "mgrip": [
      {
        "ip": "0.0.0.0",
        "access": "true"
      }, {
        "ip": "0.0.0.0",
        "access": "false"
      }
    ]
  },
  "local": {
    "admin": {
      "num": 1,
      "user": [
        {
          "username": "cyber",
          "passwd": "*****",
          "access": "true"
        }
      ]
    },
    "radius": {
      "num": 0,
      "server": []
    },
    "ldap": {
      "num": 0,
      "server": []
    },
    "session": {
      "timeout": 3
    },
    "dot1x": {
      "access": "false",
      "ident": "",
      "keypasswd": "",
      "castatus": "none",
      "certstatus": "none",
      "keystatus": "none"
    }
  }
}
```

- Show authtype of system

curl -v http://<IP>/api/security/management/authtype/ -H "token:<Token>"

- Set authtype of system

curl -X PUT http://<IP>/api/security/management/authtype/ -d '{"authtype": "localonly"}' -H "token:<Token>"

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

- Show secret of software authentication

```
curl -v http://<IP>/api/security/management/secret/ -H "token:<Token>"
```

- Set secret of software authentication

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

- Show manager IP

```
curl -v http://<IP>/api/security/management/mgrip/ -H "token:<Token>"
```

```
{  
  "mgrip": [  
    {  
      "ip": "0.0.0.0",  
      "access": "true"  
    }, {  
      "ip": "0.0.0.0",  
      "access": "false"  
    }  
  ]  
}
```

- Show secondary manager IP

```
curl -v http://<IP>/api/security/management/mgrip/2/ -H  
"token:<Token>"
```

```
{  
  "ip": "0.0.0.0",  
  "access": "false"  
}
```

- Set secondary manager IP

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

- Set manager IP by array

```
curl -X PUT http://<IP>/api/security/management/mgrip/ -d  
'{"mgrip":[{"ip":"192.168.202.11"}, {"ip":"192.168.202.44","access":"t  
rue"}]}' -H "token:<Token>"
```

(Set primary manager IP as **192.168.202.11** , Set secondary manager IP as **192.168.202.44**)

- Show Local Account

curl -v http://<IP>/api/security/local/ -H "token:<Token>"

```
{
  "admin": {
    "num": 1,
    "user": [{
      "username": "cyber",
      "passwd": "*****",
      "access": "true"
    }]
  },
  "viewer": {
    "num": 1,
    "user": [{
      "username": "device",
      "passwd": "*****",
      "access": "true"
    }]
  }
}
```

- Show Admin Account

curl -v http://<IP>/api/security/local/admin/ -H "token:<Token>"

- Show the number of admin account

curl -v http://<IP>/api/security/local/admin/num/ -H "token:<Token>"

- Add admin account

curl -X POST http://<IP>/api/security/local/admin/user/ -d
'{"username":"aaa","passwd":"aaa","access":"true"}' -H
"token:<Token>"

- Show viewer account

curl -v http://<IP>/api/security/local/viewer/ -H "token:<Token>"

- Show the number of viewer account

curl -v http://<IP>/api/security/local/viewer/num/ -H "token:<Token>"

- Add viewer account

curl -X POST http://<IP>/api/security/local/viewer/user/ -d
'{"username":"bbb","passwd":"aaa","access":"true"}' -H
"token:<Token>"

- Set secondary viewer's access as false

curl -X PUT http://<IP>/api/security/local/viewer/user/2/ -d
'{"access":"false"}' -H "token:<Token>"

- Show radius server

curl -v http://<IP>/api/security/radius/ -H "token:<Token>"

```
{
  "num": 1,
  "server": [
    {
      "hostname": "192.168.0.23",
      "secret": "*****",
      "port": 333,
      "authtype": "pap"
    }
  ]
}
```

- Show the number of radius server

curl -v http://<IP>/api/security/radius/num/ -H "token:<Token>"

- Add radius server

curl -X POST http://<IP>/api/security/radius/server/ -d
'{"hostname":"192.168.0.62","port":1812,"secret":"test","authtype":
"pap"}' -H "token:<Token>"

- ⇒ "hostname" → <STRING>
- ⇒ "port" → <NUMBER>
- ⇒ "secret" → <STRING>
- ⇒ "authtype" → "pap" 、 "chap"

- Modify primary radius server hostname

curl -X PUT http://<IP>/api/security/radius/server/1/ -d
'{"hostname":"192.168.0.111"}' -H "token:<Token>"

- Delete primary radius server

curl -X DELETE http://<IP>/api/security/radius/server/ -d '{"index":1}' -
H "token:<Token>"

- Show ldap server

curl -v http://<IP>/api/security/ldap/ -H "token:<Token>"

```
{
  "num": 1,
  "server": [
    {
      "hostname": "192.168.202.33",
      "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 http://<IP>/api/security/ldap/server/ -d

'{"type":"generic","hostname":"192.168.202.33","basedn":"dc=cyber,dc=com","userattr":"cn","port":389,"ssl":"false","auth_n_mode":"anonymous","auth_z_mode":"byattr","adminattr":"description","adminvalue":"cyber_admin"}' -H "token:<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>

- Set primary ldap server SSL as true

```
curl -X POST http://<IP>/api/security/ldap/server/1/ -d '{"ssl":"true"}' -H "token:<Token>"
```

- Set primary ldap server userattr as uid

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

- Show primary ldap server

```
curl -v http://<IP>/api/security/ldap/server/1/ -H "token:<Token>"
```

- Delete primary ldap server

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

- Show information of session

```
curl -v http://<IP>/api/security/session/ -H "token:<Token>"
```

```
{
  "timeout": 3
}
```

- Set session timeout

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

- Show information of 802.1x

```
curl -v http://<IP>/api/security/dot1x/ -H "token:<Token>"
```

```
{
  "access": "false",
  "ident": "",
  "keypasswd": "",
  "castatus": "none",
  "certstatus": "none",
  "keystatus": "none"
}
```

- Enable 802.1x

```
curl -X PUT http://<IP>/api/security/dot1x/ -d '{"access":"true"}' -H
```

“token:<Token>”

- Upload CA Certificate of 802.1x

```
curl -F upfile=@<CA file> http://<IP>/api/security/dot1x/upload/ca/ -H  
“token:<Token>”
```

- Upload Certificate of 802.1x

```
curl -F upfile=@<Cert file> http://<IP>/api/security/dot1x/upload/cert/ -H  
“token:<Token>”
```

⇒ “type” → “generic” \ “ad”

- Upload Private Key of 802.1x

```
curl -F upfile=@<Key file> http://<IP>/api/security/dot1x/upload/key/ -H  
“token:<Token>”
```

- Delete CA Certificate of 802.1x

```
curl -X DELETE http://<IP>/api/security/dot1x/ -d '{"delca":"true"}' -H  
“token:<Token>”
```

- Delete Certificate of 802.1x

```
curl -X DELETE http://<IP>/api/security/dot1x/ -d '{"delcert":"true"}' -H  
“token:<Token>”
```

- Delete Private Key of 802.1x

```
curl -X DELETE http://<IP>/api/security/dot1x/ -d '{"delkey":"true"}' -H  
“token:<Token>”
```

5. Network

curl -v http://<IP>/api/network/ -H "token:<Token>"

```
{
  "ipv4": {
    "ip": "192.168.0.115",
    "subnetmask": "255.255.255.0",
    "gateway": "192.168.0.1",
    "dns": "192.168.0.1",
    "dhcp": "true",
    "dnsfromdhcp": "true",
    "hostname": "rnc00:00:00",
    "hostnamesync": "false"
  },
  "ipv6": {
    "access": "false",
    "routercontrol": "false",
    "manual": "false",
    "linklocal": "",
    "routercontrol": "",
    "manualaddr": ""
  },
  .
  .
  .
  .
  "console": {
    "telnetaccess": "true",
    "telnetport": 23,
    "sshaccess": "true",
    "sshport": 22
  },
  "ftp": {
    "ftpaccess": "true",
    "ftpport": 21
  }
}
```

- Show IPv4 IP Address

curl -v http://<IP>/api/network/ipv4/ip/ -H "token:<Token>"

- Show IPv4 DHCP

curl -v http://<IP>/api/network/ipv4/dhcp/ -H "token:<Token>"

- Set IPv4 DHCP

curl -X PUT http://<IP>/api/network/ipv4/dhcp/ -d '{"dhcp": "false"}' -H "token:<Token>"

- Enable IPv6 access

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

- Show SNMPv1 information

```
curl -v http://<IP>/api/network/snmpv1/ -H "token:<Token>"
```

```
{
  "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 http://<IP>/api/network/snmpv1/ -d '{"access":"true"}' -H "token:<Token>"
```

- Set SNMPv1 secondary user IP as 192.168.0.201.

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

⇒ "community" → <String>

⇒ "ip" → <String>

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

- Show SNMPv3 information

```
curl -v http://<IP>/api/network/snmpv3/ -H "token:<Token>"
```

```

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

```

- Enable SNMPv3 access
curl -X PUT http://<IP>/api/network/snmpv3/ -d '{"access":"true"}' -H "token:<Token>"
- Show SNMPv3 forth user information
curl -v http://<IP>/api/network/snmpv3/user/4/ -H "token:<Token>"
- Set SNMPv3 forth user IP as 192.168.0.201
curl -X PUT http://<IP>/api/network/snmpv3/user/4/ -d '{"ip":"192.168.0.201"}' -H "token:<Token>"
- Set SNMPv3 secondary snmp user authentication protocol as md5 and authentication password as 1111111111111111.
curl -X PUT http://<IP>/api/network/snmpv3/user/2/ -d '{"authprotocol":"md5","authpasswd":"1111111111111111"}' -H "token:<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 http://<IP>/api/network/snmpv3/user/ -d
'{"user": [{"ip": "192.168.0.202"}, {"ip": "192.168.0.203"}]} -H
"token:<Token>"
```

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

- Show information of Web

```
curl -v http://<IP>/api/network/web/ -H "token:<Token>"
```

```
{
  "httpaccess": "true",
  "httpsaccess": "false",
  "httpport": 80,
  "httpsport": 443,
  "alg": [
    {
      "name": "TLS_DHE_DSS_WITH_AES_128_CBC_SHA",
      "status": "true"
    },
    {
      "name": "TLS_DHE_DSS_WITH_AES_256_CBC_SHA",
      "status": "true"
    },
    {
      "name": "TLS_DHE_RSA_WITH_AES_128_CBC_SHA",
      "status": "true"
    },
    {
      "name": "TLS_DHE_RSA_WITH_AES_256_CBC_SHA",
      "status": "true"
    },
    {
      "name": "TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA",
      "status": "true"
    },
    {
      "name": "TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA",
      "status": "true"
    },
    {
      "name": "TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA",
      "status": "true"
    },
    .
    .
    .
    .
    .
    .
    {
      "name": "TLS_RSA_WITH_AES_128_GCM_SHA256",
      "status": "true"
    },
    {
      "name": "TLS_RSA_WITH_AES_256_CBC_SHA",
      "status": "true"
    },
    {
      "name": "TLS_RSA_WITH_AES_256_GCM_SHA384",
      "status": "true"
    },
    {
      "name": "TLS_RSA_WITH_CAMELLIA_128_CBC_SHA",
      "status": "true"
    },
    {
      "name": "TLS_RSA_WITH_CAMELLIA_256_CBC_SHA",
      "status": "true"
    }
  ]
}
```

- Set HTTP port as 5000

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

- Disable TLS_DHE_DSS_WITH_AES_128_CBC_SHA algorithm of HTTPS

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

- Upload HTTPS certification

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

- Show information of console

```
curl -v http://<IP>/api/network/console/ -H "token:<Token>"
```

```
{  
  "telnetaccess": "true",  
  "telnetport": 23,  
  "sshaccess": "true",  
  "sshport": 22  
}
```

- Disable Telnet

```
curl -X PUT http://<IP>/api/network/console/ -d  
'{"telnetaccess":"false"}' -H "token:<Token>"
```

- Upload SSH hostkey

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

- Show information of FTP

```
curl -v http://<IP>/api/network/ftp/ -H "token:<Token>"
```

```
{  
  "ftpaccess": "true",  
  "ftpport": 21  
}
```

- Disable FTP

```
curl -X PUT http://<IP>/api/network/ftp/ -d '{"ftpaccess":"false"}' -H  
"token:<Token>"
```


6. Notification

curl -v http://<IP>/api/notification/ -H "token:<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 http://<IP>/api/notification/event/security/ -H "token:<Token>"

```
{
  "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"
    }, {
      "index": 63,
      "msg": "The password has been changed",
      "log": "true",
      "email": "false",
      "trap": "false",
      "syslog": "false",
      "sms": "false"
    }, {
      "index": 64,
      "msg": "Configuration file uploaded",
      "log": "true",
      "email": "false",
      "trap": "false",
      "syslog": "false",
      "sms": "false"
    }
  ]
}
```

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

```
{
  "server": "",
  "senderemail": "",
  "sendername": "",
  "auth": "false",
  "account": "",
  "passwd": "",
  "encrypt": "none"
}
```

- Set SMTP server
`curl -X PUT http://<IP>/api/notification/smtp/ -d '{"server": "smtp-mail.outlook.com"}' -H "token:<Token>"`

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

- Show e-mail recipients
`curl -v http://<IP>/api/notification/emailrcpt/ -H "token:<Token>"`

```
{
  "num": 1,
  "rcptinfo": [
    {
      "email": "test@gmail.com",
      "status": "enable"
    }
  ]
}
```

- Add e-mail recipient

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

- Delete first e-mail recipient

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

- Disable first e-mail recipient

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

⇒ "status" → "enable" 、 "disable"

⇒ "email" → <String>

- Send test message to first email recipient

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

- Show trap recipients

```
curl -v http://<IP>/api/notification/traprcpt/ -H "token:<Token>"
```

```
{
  "num": 1,
  "rcptinfo": [
    {
      "name": "Trap Name",
      "status": "true",
      "version": 1,
      "ip": "192.168.0.205",
      "community": "public"
    }
  ]
}
```

- Add SNMPv1 trap receiver

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

- Add SNMPv3 trap receiver

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

- Delete first trap receiver

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

- Set first trap receiver name as "testname"

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

⇒ "name" → <String>

⇒ "status" → "true" 、 "false"

⇒ "version" → 1 、 3

⇒ "ip" → <String>

⇒ "community" → <String>

- Send test message to first trap receiver

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

7. System

curl -v http://<IP>/api/system/ -H "token:<Token>"

```
{
  "sn": "NA",
  "modelname": "RMCARD400",
  "mac": "00-0C-15-05-7B-77",
  "hwversion": "1.0",
  "kernelversion": "1.0.1",
  "updatedate": "12/26/2023 16:52:20",
  "fwversion": "1.0.4 (Beta Version : 0.6) "
```

- Show model name

curl -v http://<IP>/api/system/modelname/ -H "token:<Token>"

- Reboot system

curl -X POST http://<IP>/api/system/reboot/ -d '{"reboot":"true"}' -H "token:<Token>"

- Reset system

curl -X POST http://<IP>/api/system/reset/ -d '{"reset":"true"}' -H "token:<Token>"

- Reset system (TCP/IP Settings Reserved)

curl -X POST http://<IP>/api/system/resetnotcpip/ -d '{"resetnotcpip":"true"}' -H "token:<Token>"

- Upload restore configuration file

curl -F upfile=@<Restore file> http://<IP>/api/system/restore/upload/ -H "token:<Token>"

- Download save configuration file

curl http://<IP>/api/system/restore/download/ --output <Download File Name> -H "token:<Token>"

- Download diagnostic information file

curl http://<IP>/api/system/diagnoinfo/download/ --output <Download File Name> -H "token:<Token>"

- Firmware update

curl -F upfile=@<FW file> http://<IP>/api/system/firmware/upload/ -H "token:<Token>"

8. UPS Status

curl -v http://<IP>/api/upsstatus/ -H "token:<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)

9. UPS Information

curl -v http://<IP>/api/upsinfo/ -H "token:<Token>"

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

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

10. UPS Configuration

curl -v http://<IP>/api/upsconfig/ -H "token:<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 http://<IP>/api/upsconfig/ suppliedpwr/ -H "token:<Token>"

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

- Set supplied power as 120V

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

- Show utility power failure condition of UPS configuration

curl -v http://<IP>/api/upsconfig/ suppliedpwr/pwrfailcondi/ -H "token:<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 http://<IP>/api/upsconfig/pwrfailcondi/ -d '{"freqtol":3}' -H "token:<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 http://<IP>/api/upsconfig/suppliedpwr/operation/ -H "token:<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 http://<IP>/api/upsconfig/operation/exdays/ -d
'{"exdays":[{"status":"true"}, {"status":"true"}]} -H "token:<Token>"
```

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

- Show bypass of UPS configuration

```
curl -v http://<IP>/api/upsconfig/bypass/ -H "token:<Token>"
```

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

- Set bypass condition as "Check Volt Only"

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

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

- Show power restore of UPS configuration

```
curl -v http://<IP>/api/upsconfig/pwrrestore/ -H "token:<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 http://<IP>/api/upsconfig/pwrrestore/ -d '{"autorestore":"false"}' -H "token:<Token>"
```

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

- Show battery of UPS configuration

```
curl -v http://<IP>/api/upsconfig/battery/ -H "token:<Token>"
```

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

- Set low battery threshold as 30%

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

- Set periodical battery test as 2 weeks

```
curl -X PUT http://<IP>/api/upsconfig/battery/ -d '{"periodtest":20160}' -H "token:<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 http://<IP>/api/upsconfig/battery/ -H "token:<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 http://<IP>/api/upsconfig/system/ -d '{"coldstart":"false"}'
-H "token:<Token>"

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

11. UPS Master Switch

```
curl -v http://<IP>/api/upsswitch/ -H "token:<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 http://<IP>/api/upsswitch/ -d '{"turnon":"true"}' -H "token:<Token>"
```

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

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

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

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

12. UPS Bank

```
curl -v http://<IP>/api/upsbank/ -H "token:<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 http://<IP>/api/upsbank/bank/2/ -d '{"status":"off"}' -H "token:<Token>"
```

- Set first outlet name of bank 1

```
curl -X PUT http://<IP>/api/upsbank/bank/1/outlet/1/ -d '{"name":"test_outlet_name"}' -H "token:<Token>"
```

13. UPS Diagnostics

```
curl -v http://<IP>/api/upsdiagno/ -H "token:<Token>"
```

```
{
  "ltresult":      "pass",
  "ltdate":       "Dec-26-23 ",
  "lesresult":    "",
  "lesdate":     ""
}
```

- Execute the battery test

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

- Execute runtime estimation

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

- Abort runtime estimation

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


14. UPS Schedule

```
curl -v http://<IP>/api/upssche/ -H "token:<Token>"
```

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

- Show first schedule

```
curl -v http://<IP>/api/upssche/schedule/1/ -H "token:<Token>"
```

15. UPS Wake on lan

```
curl -v http://<IP>/api/upswol/ -H "token:<Token>"
```

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

16. UPS Event Log

```
curl -v http://<IP>/api/upsevent/ -H "token:<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 http://<IP>/api/upsevent/event/ -d '{"nextpage":"true"}' -H
"token:<Token>"
```

- Review events

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

- Clear all events

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

- Download event log file

```
curl http://<IP>/api/upsevent/event/download/ --output <Download File
Name> -H "token:<Token>"
```

17. UPS Record Data

`curl -v http://<IP>/api/upsrec/ -H "token:<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 http://<IP>/api/upsrec/rec/ -d '{"nextpage":"true"}' -H "token:<Token>"`

- Review records

`curl -X PUT http://<IP>/api/upsrec/rec/ -d '{"review":"true"}' -H "token:<Token>"`

- Clear all records

`curl -X PUT http://<IP>/api/upsrec/rec/ -d '{"reset":"true"}' -H "token:<Token>"`

- Download record data file

`curl http://<IP>/api/upsevent/event/download/ --output <Download File Name> -H "token:<Token>"`

- Set accessory basic

```
curl -X PUT http://<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  
"token:<Token>"
```

- Set accessory contact

```
curl -X PUT http://<IP>/api/accessory/env/device/2/config/ -d  
'{"contact4name":"conname4","contact4state":"close"}' -H  
"token:<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-E000097-00