

API 中心管理模块

POST /system/init

说明

初始化中心。

先初始化 超级管理员 和 系统管理员两个用户，再调用此接口才能初始化中心成功。

如果中心已经被初始化，不允许再初始化。

request

```
{}
```

response

```
{
  "code": "success",
  "msg": "success",
  "data": {}
}
```

GET /system/info

说明

列出中心基本信息，如设备厂商、设备型号、序列号和初始化状态等

response

```
{
  "code": "success",
  "msg": "success",
  "data": {
    "device_vendor": "<string>", //
    "device_model": "<string>", //
    "series_number": "<string>", //
    "product_version": "<string>", //
    "software_version": "<string>", //
    "init_timestamp": "<integer>", //epoch time ms 1722306732717 -1
    "system_time": "<integer>", //epoch time ms 1722306732717
    "uptime": "<integer>" //epoch time ms 6732717
  }
}
```

GET /system/status

说明

获取一段时间内中心的状态监控数据

监控间隔为1秒时，JSON形式的 30 天状态数据大小约为111MB。为了减少处理和传输的压力，请求数据时需要指定"采样数"以限制要获取的最大数据条数。

每两条数据间的数据条数将尽可能一致。

包含时间段开始和结束附近的数据。

请求

请求类型：Query

| 名称 | 数据类型 | 必填 | 说明 |
|-----------------|---------|------|------------------------------------|
| type | string | true | 监控指标，取值见下方列表 |
| sample_num | integer | true | 前端需要返回的数据点的数目 |
| start_timestamp | integer | true | epoch time ms 示例： 1722305732717 |
| end_timestamp | integer | true | epoch time ms 示例： 1722305732717 |

type取值：

| 监控指标名称 | 说明 |
|---------------------------|-------------------|
| cpu_used_percent | CPU 占用百分比，% |
| memory_used | 内存占用字节数，Byte |
| storage_used | 存储占用字节数，Byte |
| tx_traffic_byte_rate | 网卡发送数据速率，Byte / s |
| rx_traffic_byte_rate | 网卡接收数据速率，Byte / s |
| tx_history_traffic_volume | 网卡发送数据总量，Byte |

| | |
|-------------------------------|----------------|
| rx_history_traffic_volum e | 网卡接收数据总量, Byte |
|-------------------------------|----------------|

response

```
{
  "code": "success",
  "msg": "success",
  "data": {
    // type: cpu_used_percent
    "history": [
      {
        "timestamp": "<integer>", //epoch time ms 1722306732717
        "value": "<integer>" //
      },
      ...
    ],

    // type: memory_used
    "total": "<integer>",
    "history": [
      {
        "timestamp": "<integer>", //epoch time ms 1722306732717
        "value": "<integer>" //
      },
      ...
    ],

    // type: storage_used
    "total": "<integer>",
    "history": [
      {
        "timestamp": "<integer>", //epoch time ms 1722306732717
        "value": "<integer>" //
      },
      ...
    ],

    // type: tx_traffic_byte_rate / rx_traffic_byte_rate
    "history": [
      {
        "timestamp": "<integer>", //epoch time ms 1722306732717
        "value": "<integer>" //
      },
      ...
    ],

    // type: tx_history_traffic_volum / rx_history_traffic_volum
    "history": [
      {
        "timestamp": "<integer>", //epoch time ms 1722306732717
        "value": "<integer>" //
      },
      ...
    ],
  }
}
```

GET /system/service

说明

获取某个服务的配置，服务包含 Web、OpenVPN、SNMP 和 LDAP。

OpenVPN 和 SNMP 是部署于中心所在服务器上的第三方程序。

用户鉴权时，中心作为客户端访问远程 LDAP 服务器。

query

```
{
  "service_name": "<string>" //, : "web" "openvpn" "snmp" "ldap"
}
```

response

注意：data中的数据个根据service_name不同，返回参数不一样

```
{
  "code": "success",
  "msg": "success",
  "data": {
    //service_name: web
    "port": "<integer>"

    //service_name: openvpn
    "port": "<integer>"

    //service_name: snmp
    "enable": "<bool>"

    //service_name: ldap
    "host": "<string>", // LDAP ip,ip_reg
    "port": "<integer>", // LDAP ,(0~65535)
    "uid": "<string>", // LDAP
    "bind_dn": "<string>", // LDAP DN
    "password": "<string>", // LDAP ,
    "user_filter": "<string>", // LDAP ,
    "base": "<string>" //
  }
}
```

PUT /system/service

说明

更新某个服务的配置

request

```
{
  "web_config":{
    "port": "<integer>"
  }
}

{
  "openvpn_config": {
    "port": "<integer>"
  }
}

{
  "snmp_config": {
    "enable": "<bool>"
  }
}

{
  "ldap_config": {
    "host": "<string>",      // LDAP ip,ip_reg
    "port": "<integer>",    // LDAP ,(0~65535)
    "uid": "<string>",      // LDAP
    "bind_dn": "<string>",  // LDAP DN
    "password": "<string>", // LDAP ,
    "user_filter": "<string>", // LDAP ,
    "base": "<string>"      //
  }
}
```

response

```
{
  "code": "success",
  "msg": "success",
  "data": {}
}
```

GET /system/network

说明

列出网络设备信息，例如各网卡 IP、DNS 服务器列表、公网 IP 和域名

response

```

{
  "code": "success",
  "msg": "success",
  "data": {
    "nics": [
      {
        "name": "<string>", //
        "ips": [
        ]
        "role": "<string>" // ttn/user/unset
      },
      ...
    ],
    "bonds": [
      {
        "name": "<string>",
        "mode": "<string>", //mode=balance-rr -
        "ips": [
          {
            "is_ipv4": "<bool>", //ipv4
            "cidr": "<string>" //ipcidr
          },
          ...
        ],
        "role": "<string>" // ttn/user/unset
        "slaves": [
          "<string>",
          ....
        ]
      },
      ...
    ],
    "gateway": {
      "ip": "<string>",
      "nic_name": "<string>"
    },
    "dns_ips": [ //dnsip
      "<string>",
      ...
    ],
    "domain_names": [ //
      "<string>",
      ...
    ],
    "public_ips": [ //IP
      "<string>",
      ...
    ]
  }
}

```

PUT /system/network

说明:

更新网络设备信息

request

```

{
  "nics": [
    {
      "name": "<string>", //
      "ips": [
        {
          "is_ipv4": "<bool>", //ipv4
          "cidr": "<string>" //ipcidr
        },
        ...
      ],
      "role": "<string>" // ttn/user/unset
    },
    ...
  ],
  "bonds": [
    {
      "name": "<string>",
      "mode": "<string>", //mode=balance-rr -
      "ips": [
        {
          "is_ipv4": "<bool>", //ipv4
          "cidr": "<string>" //ipcidr
        },
        ...
      ],
      "slaves": [
        "<string>",
        ....
      ],
      "role": "<string>" // ttn/user/unset
    },
    ...
  ],
  "gateway": {
    "ip": "<string>",
    "nic_name": "<string>"
  }
}

{
  "dns_ips": [ //dnsip
    "<string>",
    ...
  ]
}

{
  "domain_names": [ //
    "<string>",
    ...
  ],
  "public_ips": [ //IP
    "<string>",
    ...
  ]
}

```

response

```
{
  "code": "success",
  "msg": "success",
  "data": {}
}
```

GET /system/monitor

说明:

获取监控配置，例如包含中心多长时间采集数据、采集数据的大小以及警告条件。

采集数据存放在数据库中，当采集数据大于data_volume_limit参数时候，中心系统会对采集数据表进行大小对半删除处理

response

```
{
  "code": "success",
  "msg": "success",
  "data": {
    "center_interval": "<integer>", // (cpu, ), 1000 default: null
    "endpoint_interval": "<integer>", // (cpu, ), 1000 default: null
    "data_volume_limit": "<integer>", // 49152 default: null
    "center_alarm_config": {
      "cpu_used_percent": {
        "threshold": "<integer>", // 3030% default: null, 1100
        "duration": "<integer>" // default: null, 1000
      },
      "memory_used_volume": {
        "threshold": "<integer>", // byte default: null, 1
        "duration": "<integer>" // default: null, 1000
      },
      "memory_used_percent": {
        "threshold": "<integer>", // 3030% default: null, 1100
        "duration": "<integer>" // default: null, 1000
      },
      "storage_used_volume": {
        "threshold": "<integer>", // byte default: null, 1
        "duration": "<integer>" // default: null, 1000
      },
      "storage_used_percent": {
        "threshold": "<integer>", // 3030% default: null, 1100
        "duration": "<integer>" // default: null, 1000
      },
      "traffic_rate": {
        "tx_bytes": "<integer>", // byte/s default: null, 1
        "rx_bytes": "<integer>", // byte/s default: null, 1
        "tx_duration": "<integer>", // default: null, 1000
        "rx_duration": "<integer>" // default: null, 1000
      },
      "traffic_volume": {
        "tx_bytes": "<integer>", // byte default: null, 1
        "rx_bytes": "<integer>", // byte default: null, 1
        "interval": "<integer>" // default: null, 3600000
      }
    },
    "endpoint_alarm_config": {...} // key center_alarm_config
  }
}
```


PUT /system/monitor

说明:

更新监控配置

采集数据存放在数据库中，当采集数据大于data_volume_limit参数时候，中心系统会对采集数据表进行大小对半删除处理

request: (根据用户选择更改类型不同，从以下选择对应的类型发送)

```
{
  "monitor_config": {
    "center_interval": "<integer>",    //(cpu, ),1000default: null
    "endpoint_interval": "<integer>",  //(cpu, ),1000default: null
    "data_volume_limit": "<integer>"   // 49152 default: null
  }
}

{
  "center_alarm_config": {
    "cpu_used_percent": {
      "threshold": "<integer>", // 3030%default: null,1100
      "duration": "<integer>"   //default: null1000
    },
    "memory_used_volume": {
      "threshold": "<integer>", //bytedefault: null1
      "duration": "<integer>"   //default: null1000
    },
    "memory_used_percent": {
      "threshold": "<integer>", // 3030%default: null1100
      "duration": "<integer>"   //default: null1000
    },
    "storage_used_volume": {
      "threshold": "<integer>", //bytedefault: null1
      "duration": "<integer>"   //default: null1000
    },
    "storage_used_percent": {
      "threshold": "<integer>", // 3030%default: null1100
      "duration": "<integer>"   //default: null1000
    },
    "traffic_rate": {
      "tx_bytes": "<integer>",    //byte/sdefault: null1
      "rx_bytes": "<integer>",    //byte/sdefault: null1
      "tx_duration": "<integer>", //default: null1000
      "rx_duration": "<integer>"  //default: null1000
    },
    "traffic_volume": {
      "tx_bytes": "<integer>",    //bytedefault: null1
      "rx_bytes": "<integer>",    //bytedefault: null1
      "interval": "<integer>"     //default: null3600000
    }
  }
}

{
  "endpoint_alarm_config": { ... }    // key center_alarm_config
}
```

response

```
{  
  "code": "success",  
  "msg": "success",  
  "data": {}  
}
```