OneNet接入方案与RESTfulAPI接口

欢迎访问设备云门户网站<http://open.iot.10086.cn/>注册用户，获取最新文档。

|  |  |  |  |
| --- | --- | --- | --- |
| **版本号** | **修订日期** | **修订内容** | **说明** |
| V1.1 | 2014.10.30 | EDP增加存储数据报文；  REST API 增加历史数据查询接口，用于上报数据点，或者上报的同时转发数据点。 |  |
| V1.1.1 | 2014/11/3 | EDP增加消息类型9，以支持对存储数据的确认。 |  |
| V1.1.2 | 2015/01/10 | 1、修改3.5.3 AB接口，增加获取每个数据流当前值。  2、数据点添加时若无数据流将自动添加数据流。 |  |
|  | 2015/01/15 | 1. 修改3.5.1 ，新增数据流返回uuid，唯一标识该数据流； 2. 修改3.5.1，增加三种复合数据流的添加； 3. 修改触发器相关操作接口。一个触发器可同时监控不同设备的多个数据流。 4. 修改触发器一些条件的设定方式。 | 预计上线日期  2015年4月15日 |

目录

[OneNet接入方案与RESTfulAPI接口 1](#_Toc415235576)

[1 功能 2](#_Toc415235577)

[2 设备与业务接入模式 3](#_Toc415235578)

[2.1 数据存储 3](#_Toc415235579)

[2.2 数据存储并转发 3](#_Toc415235580)

[2.3 数据透传 4](#_Toc415235581)

[2.4 无控制命令下发模型 4](#_Toc415235582)

[3 资源模型 5](#_Toc415235583)

[4 REST API 6](#_Toc415235584)

[4.1 设计原则 6](#_Toc415235585)

[4.2 用户（user） 6](#_Toc415235586)

[4.2.1 新增 6](#_Toc415235587)

[4.2.2 查看 7](#_Toc415235588)

[4.3 设备（device） 7](#_Toc415235589)

[4.3.1 新增 7](#_Toc415235590)

[4.3.2 更新 8](#_Toc415235591)

[4.3.3 查看 9](#_Toc415235592)

[4.3.4 删除 11](#_Toc415235593)

[4.4 数据流（datastream） 12](#_Toc415235594)

[4.4.1 新增 12](#_Toc415235595)

[4.4.2 更新 13](#_Toc415235596)

[4.4.3 查看 14](#_Toc415235597)

[4.4.4 删除 15](#_Toc415235598)

[4.5 数据点（datapoint） 15](#_Toc415235599)

[4.5.1 新增 15](#_Toc415235600)

[4.5.2 更新 16](#_Toc415235601)

[4.5.3 查看 16](#_Toc415235602)

[4.5.4 删除 18](#_Toc415235603)

[4.6 触发器（trigger） 18](#_Toc415235604)

[4.6.1 新增 18](#_Toc415235605)

[4.6.2 更新 20](#_Toc415235606)

[4.6.3 查看 20](#_Toc415235607)

[4.6.4 删除 21](#_Toc415235608)

[4.7 API 权限(key) 21](#_Toc415235609)

[4.7.1 新增 21](#_Toc415235610)

[4.7.2 更新 22](#_Toc415235611)

[4.7.3 查看 22](#_Toc415235612)

[4.7.4 删除 23](#_Toc415235613)

[4.8 二进制数据（bindata） 23](#_Toc415235614)

[4.8.1 新增 23](#_Toc415235615)

[4.8.2 查看 24](#_Toc415235616)

[4.8.3 删除 24](#_Toc415235617)

[4.9 其他 25](#_Toc415235618)

[4.9.1 传感器触发（异常事件内容） 25](#_Toc415235619)

[4.9.2 发送数据到EDP设备 25](#_Toc415235620)

[4.9.3 RestAPI 日志 26](#_Toc415235621)

[4.9.4 历史数据查询 27](#_Toc415235622)

# 功能

* 资源操作：用户、设备、数据流、数据点、触发器和api-key；
* 海量终端接入鉴权；
* 实时消息路由和转发；
* 时间序列化数据存储；
* 数据监控、告警触发；

# 设备与业务接入模式

## 数据存储

**设备层：**实现EDP协议，用于上报业务数据点到OnetNet，接收业务层的发送的控制命令，返回命令响应到业务层。

**业务应用层：**创建EDP虚拟设备，用于下发控制命令到设备，并接收设备响应数据；通过HTTP 协议的RESTful API操作设备云提供的资源（设备、数据点等资源的增删查改）。



适用场景：在充分分析业务数据模型的基础上，认为OneNet提供的设备-数据流-数据点模型适合业务数据存储。优先推荐新业务使用该模式。

## 数据存储并转发

**设备层：**实现EDP协议，用于上报业务数据点到OnetNet，接收业务层的发送的控制命令，返回命令响应到业务层。

**业务应用层：**创建EDP虚拟设备，用于下发控制命令到设备，并接收设备响应数据，实时接收设备上报的业务数据（该数据由设备层产生，发送到OneNet平台进行保存后，再通过EDP转发到业务层）；通过HTTP 协议的RESTful API操作设备云提供的资源（设备、数据点等资源的增删查改）。



适用场景：当设备上报的数据需要实时分析和统计，而OneNet又暂时缺乏该复杂分析的能力时，可以将数据保存的同时，转发一份到应用层，由应用层进行综合分析。

## 数据透传

**设备层：**实现EDP协议，用于接收业务层的发送的控制命令，返回命令响应到业务层。

**业务应用层：**创建EDP虚拟设备，用于下发控制命令到设备，并接收设备响应数据（OneNet平台负责业务与设备间消息中转，对传输内容不解析和存储）；通过HTTP 协议的RESTful API操作设备云提供的资源（设备、数据点等资源的增删查改）。



适用场景：业务数据格式有特殊需求，不能按设备-数据流-数据点进行划分；或者终端上报的数据不能直接存储，需要经过加工后再存储。

## 无控制命令下发模型

**设备层：**通过HTTP 协议的RESTful API上报业务数据点。

**业务应用层：**通过HTTP 协议的RESTful API操作设备云提供的资源（设备、数据点等资源的增删查改）。



适用场景：终端不需要实时接收控制命令的业务模型，终端只是采集数据并上报到业务平台。因为RESTful接口不能对设备进行实时的消息下达。

# 资源模型

平台提供设备（device）、数据流（datastream）、数据点（datapoint）、触发器（trigger）、API key等资源，通过REST API 可以对平台进行操作增删查改操作。



* 每个平台用户，可以创建自己的设备列表，设定设备相关属性；
* 每个设备下，可以创建多个数据流；
* 数据流是某一类按时间顺序存储的数据点；
* 数据点则是以时间戳为key，任意json数据类型为value的key-value对；
* 针对每个数据流，可设定对数据点进行监控的触发器；
* Key是用来规定用户是否具有操作相关资源的权限，细化到数据流级别；

# REST API

REST API 基于HTTP 协议（详见http://www.w3.org/Protocols/HTTP/1.0/spec.html）和json数据格式（详见http://www.json.org/json-zh.html），适合平台与平台之间数据对接，或使用短连接上报终端数据。

<API\_ADDRESS>外网运营地址：api.heclouds.com端口80 或jjfarfapi.heclouds.com端口80

## 设计原则

* 使用标准HTTP方法实现资源CURD操作；
* 采用json作为API输入输出；
* 以json输出错误信息。

|  |  |
| --- | --- |
| HTTP 方法 | 描述 |
| POST | 新增资源信息 |
| PUT | 更新资源信息 |
| GET | 查看资源信息 |
| DELETE | 删除资源 |

支持的返回码列表：

|  |  |  |
| --- | --- | --- |
| **HTTP返回码** | **含义** | **详细描述** |
| 200 | OK | 在HTTP body中以json格式指明详细成功或错误信息，例如：  **{**  **"errno": 0,**  **"error":“succ”**  **}**  成功、失败指示，主要以errno是否为零来判断。 |
| 400 | Bad Request |
| 401 | Not Authorized |
| 403 | Forbidden |
| 404 | Not Found |
| 406 | Not Acceptable |
| 422 | Unprocessable Entity |
| 500 | Internal Server Error |
| 503 | No server error |

## HTTP 原始报文格式

例1，自己封装HTTP报文，新增设备，格式如下，其中\r\n是两个字符（ASCII码里面的回车换行），长度32表示两个连续\r\n\r\n后数据的长度：

**POST /devices HTTP/1.1\r\n  
api-key:<your\_api\_key>\r\n  
Host:api.heclouds.com\r\n  
Content-Length: 32\r\n  
\r\n  
{"title":"898602B2221430000029"}**

例2，看某个设备的详细情况，格式如下，

**GET /devices/<your\_device\_id> HTTP/1.1\r\n  
api-key:<your\_api\_key>\r\n  
Host:api.heclouds.com\r\n  
\r\n**

## 用户（user）

### 新增

通过设备云门户网站[www.heclouds.com](http://www.heclouds.com/)注册用户，在“设置“页面中查看user\_id 和master\_key。

### 查看

|  |  |
| --- | --- |
| HTTP方法 | GET |
| URL | http://<API\_ADDRESS>/user |
| URL参数 |  |
| HTTP头部 | api-key:xxxx-ffff-zzzzz  （注：只能为master-key） |
| 请求返回 | **{**  **"errno": 0,**  **"error":“succ” ,**  **"data":**  **{**  **"id":23028,//整型**  **"name":"xiaowang",**  **"email":"23929@qq.com"，**  **"binary":[//用户关联的图像或二进制数据**  **{**  **“index”:”FJWOPN9023899”, //二进制数据索引**  **“at”:”2014-10-23 20:22:22”,//上传时间**  **“size”:2333(字节),//二进制数据大小**  **”type”:”binary|png|gif|jpeg”,//数据类型**  **”width”:220,//图片宽度**  **”height”:300,//图片高度**  **”desc”:字符串或json对象//根据上传时的desc参数确定**  **}**  **]**  **}**  **}** |

## 设备（device）

### 新增

注：DTU类型设备，title填写为卡号，auth\_info填写DTU中配置的卡号（phon）和密码（svrpwd），route\_to填写该DTU设备采集数据后默认的转发地址（某个虚拟设备的ID号）；

对于采用激活方式登录的EDP终端，需要在新增设备时，将activate\_code预先定义和注册好。

|  |  |
| --- | --- |
| HTTP方法 | POST |
| URL | http://<API\_ADDRESS> /devices |
| HTTP头部 | api-key:xxxx-ffff-zzzzz |
| HTTP内容 | **{**  **"title":"my device1",//用户范围内唯一**  **"desc":"some description,ex:url",//可选**  **"tags":[//可选**  **"Tag1",**  **"Tag2"**  **],**  **"url":"http://1.2.3.4/abc/def",//可选 http transfer的默认接收地址**  **"idsn":"136264521",//可选**  **"location":{//可选**  **"ele":370000,**  **"lat":17.609997,**  **"lon":177.03403**  **},**  **"private":"true",**  **"route\_to":"2113",//socket接口发送数据的默认路由地址（设备ID号）**  **"auth\_info":{…},//可选,用于socket接口鉴权,** DTU类型设备,该字段需包含卡号（phone）和密码（svrpwd）  **"interval":60,//可选,modbus TCP 类设备，采集间隔，秒**  **"activate\_code":{…},//可选,用于socket接口激活**  **"other":{…}//可选**  **}** |
| 请求返回 | **{**  **"errno": 0,**  **"error":“succ”,**  **"data":**  **{**  **"device\_id":"233444"//平台分配唯一ID**  **}**  **}** |

### 更新

根据更新参数，全量设置对应项目。

|  |  |
| --- | --- |
| HTTP方法 | PUT |
| URL | http://<API\_ADDRESS>/ devices /<device\_id> |
| HTTP头部 | api-key:xxxx-ffff-zzzzz |
| HTTP内容 | **{**  **"title":"my device1",**  **"desc":"some description,ex:url",**  **"tags":["Tag1"，"Tag2"],**  **"private":"true",**  **"location":{**  **"ele":370000,**  **"lat":17.609991828964787,**  **"lon":177.03402996826173**  **},**  **"route\_to":"2113",**  **"auth\_info":{…},**  **"interval":60,//可选,modbus TCP 类设备，采集间隔，秒**  **"other":{}**  **}** |
| 请求返回 | **{**  **"errno": 0,**  **"error":“succ”,**  **}** |

### 查看

a单个查看

|  |  |
| --- | --- |
| HTTP方法 | GET |
| URL | http://<API\_ADDRESS>/devices/<device\_id> |
| HTTP头部 | api-key:xxxx-ffff-zzzzz |
| 请求返回 | **{**  **"errno": 0,**  **"error":“succ”,**  **"data":**  **{**  **"id":"device\_id",**  **"online":true|false,**  **"title":"my device1",**  **"desc":"some description,ex:url",**  **"create\_time":"xx-xx-xx 10:22:22",**  **"activate\_time":"xxxx-xx-xx 12:22:22",**  **"binary":[//设备关联的图像或二进制数据**  **{**  **“index”:”FJWOPN9023899”, //二进制数据索引**  **“at”:”2014-10-23 20:22:22”,//上传时间**  **“size”:2333(字节),//二进制数据大小**  **”type”:”binary|png|gif|jpeg”,//数据类型**  **”width”:220,//图片宽度**  **”height”:300,//图片高度**  **”desc”:字符串或json对象//根据上传时的desc参数确定**  **}]**  **"tags":["Tag1","Tag2"],**  **"location":{**  **"ele":370000,**  **"lat":17.609991828964787,**  **"lon":177.03402996826173**  **},**  **"online":true|false,**  **"private":true|false,**  **"route\_to":"2113",**  **"auth\_info":{…},**  **"other":{}**  **"keys":[**  **{"title":"device-auto-key","key":"2JO3B12O1"},**  **{"title":"key-name2","key":"DFWfsfw1wfDF233"},**  **]，**  **"datastreams":[**  **{//数据流**  **"id":"datastream\_id1”,//设备范围内唯一**  **"~~title":"my device1",~~**  **~~"dev\_id":dev\_id1,~~**  **"tags":[**  **"Tag1"**  **"Tag2"**  **],**  **"unit":"celsius",**  **"unit\_symbol":"C",**  **"type":"num|str|bin|json|other",**  **~~"current\_value":{…}~~,**  **~~"update\_at":"xx-xx-xx"~~**  **},{…}…**  **]//end data\_streams**  **}**  **}** |
|  |  |

b批量查看

|  |  |
| --- | --- |
| HTTP方法 | GET |
| URL | http://api.onenet.com/devices |
| URL参数 | **key\_words = xxbbcc,//可选**  **tag = xxx //可选**  **online = true|false //可选**  **private = true|false //可选**  **page = 1 //**指定页码, **可选**  **per\_page = 30 //**指定每页输出设备个数,**可选,**默认30，最多100  使用示例：  http://api.onenet.com/devices?page=3&per\_page=100 |
| HTTP头部 | api-key:xxxx-ffff-zzzzz |
| 请求返回 | **{**  **"errno": 0,**  **"error":“succ”,**  **"data":**  **{**  **"total\_count":600,**  **"per\_page":30,**  **"page":1,**  **"devices":[{**  **"id":"dev\_id",**  **"title":"my device1",**  **"online":ture|false,**  **"create\_time":"2013-02-11 12:22:33"**  **},{…}]**  **}**  **}** |

### 删除

|  |  |
| --- | --- |
| HTTP方法 | DELETE |
| URL | http://<API\_ADDRESS>/ devices /<device\_id> |
| HTTP头部 | api-key:xxxx-ffff-zzzzz |
| HTTP内容 | **无** |
| 请求返回 | **{**  **"errno": 0,**  **"error":“succ”**  **}** |

## 数据流（datastream）

### 新增

|  |  |
| --- | --- |
| HTTP方法 | POST |
| URL | http://<API\_ADDRESS>/devices/<device\_id>/datastreams |
| HTTP头部 | api-key:xxxx-ffff-zzzzz |
| HTTP内容 | **{**  **"id":"datastream\_id1”,//设备范围内唯一**  **~~"title":"my device1",~~**  **~~"dev\_id":dev\_id1,~~**  **//以下参数可选**  **"tags":[**  **"Tag1"**  **"Tag2"**  **],**  **"unit":"celsius",**  **"unit\_symbol":"C",**  **//复合数据流填写**  **"combine"://对象**  **----------- combine type == 1--------------**  **//从某时刻开始，每个周期累加，children必须填写某个单一数据流**  **{**  **"type":1,// int 类型**  **"cycle":"day",//累加周期，day：1天，week：一周，month：月，quarter:一个季度,year:一年**  **"begin\_day":1，//从一周、一月等的第几天开始累加，周期为天时该字段不用设置**  **"children":["datastream1\_uuid”]//必须填1个**  **}**  **----------- combine type== 2--------------**  **//计算children中指定的唯一数据流最近两数据差值，children必须填写某个单一数据流**  **{**  **"type":2,**  **"children":["datastream1\_uuid”]//必须填1个**  **}**  **------------ combine type == 3-------------**  **//根据children中列的数据流最新数据，按公式计算; v0 … vn 代表children数组中第0到n数据流的最新值**  **{**  **"type":3,**  **"op\_formula":"(v0+v1)\*v2",**  **"children":[**  **"datastream1\_uuid”,**  **"datastream2\_uuid”,**  **"datastream3\_uuid”**  **]**  **}**  **-------------------------**  **//以下参数modbus类型设备填写**  **"cmd":"0003000000184411",//MODBUS命令，16进制字节字符串**  **"interval":60,//采集间隔，秒**  **"formula":"(A0+A1)\*A2",//最终结果处理方法，寄存器地址从A0开始,多个公式用分号间隔**  **}** |
| 请求返回 | **{**  **"errno": 0,**  **"error":“succ”,**  **"data":**  **{**  **"ds\_uuid":"FWFW-WFWW-EWFW-WW"//平台分配数据流唯一ID**  **}**  **}** |

### 更新

根据更新参数，全量设置对应项目。

|  |  |
| --- | --- |
| HTTP方法 | PUT |
| URL | http://<API\_ADDRESS>/devices/<device\_id>/datastreams/<datastream\_id> |
| HTTP头部 | api-key:xxxx-ffff-zzzzz |
| HTTP内容 | **{**  **"tags":[**  **"Tag1"**  **"Tag2"**  **],**  **"unit":"celsius",**  **"unit\_symbol":"C",**  **//符合数据流属性(必须将该类型相关属性一并修改)**  **"combine":{},**  //以下参数modbus类型设备可选填写  **"cmd":"0003000000184411",//MODBUS命令，16进制字节字符串**  **"interval":60,//采集间隔，秒**  **"formula":"(A0+A1)\*A2",//最终结果处理方法，寄存器地址从A0开始**  **}** |
| 请求返回 | **{**  **"errno": 0,**  **"error":“succ”,**  **}** |

### 查看

A查看某设备单个数据流信息：

|  |  |
| --- | --- |
| HTTP方法 | GET |
| URL | http://<API\_ADDRESS>/devices/<device\_id>/datastreams/<datastream\_id> |
| HTTP头部 | api-key:xxxx-ffff-zzzzz |
| 请求返回 | **{**  **"errno": 0,**  **"error":“succ”,**  **"data":**  **{//数据流信息**  **"id":"datastream\_id1”,//设备范围内唯一**  **"uuid":"EFEW-EFW-EFW-WEFE”,//平台范围内唯一**  **"tags":[**  **"Tag1"**  **"Tag2"**  **],**  **"unit":"celsius",**  **"unit\_symbol":"C",**  **"create\_time":"xxxx-xx-xx 10:22:22",**  **"combine":{…},**  **"current\_value":{…},**  **"update\_at":"xx-xx-xx"**  **}**  **}** |

B查看某设备多个数据流信息：

|  |  |
| --- | --- |
| HTTP方法 | GET |
| URL | http://<API\_ADDRESS>/devices/<device\_id>/datastreams |
| URL参数 | **datastream\_ids = xxbbcc,//可选,多个用逗号分隔**  使用示例：  http://api.onenet.com/devices/1234/datastreams?datastream\_ids=temp,gps |
| HTTP头部 | api-key:xxxx-ffff-zzzzz |
| 请求返回 | **{**  **"errno": 0,**  **"error":“succ”,**  **"data":**  **[{//数据流信息**  **"id":"datastream\_id1”,//设备范围内唯一**  **"uuid":"EFEW-EFW-EFW-WEFE”,//平台范围内唯一**  **"tags":[**  **"Tag1"**  **"Tag2"**  **],**  **"unit":"celsius",**  **"unit\_symbol":"C",**  **"create\_time":"xxxx-xx-xx 10:22:22",**  **"combine":{…},**  **"current\_value":{…},**  **"update\_at":"xx-xx-xx"**  **}]**  **}** |

### 删除

|  |  |
| --- | --- |
| HTTP方法 | DELETE |
| URL | http://<API\_ADDRESS>/ devices /<device\_id>/datastreams/<datastream\_id> |
| HTTP头部 | api-key:xxxx-ffff-zzzzz |
| HTTP内容 | **无** |
| 请求返回 | **{**  **"errno": 0,**  **"error":“succ”,**  **}** |

## 数据点（datapoint）

### 新增

|  |  |
| --- | --- |
| HTTP方法 | POST |
| URL | *http://*<API\_ADDRESS>*/devices/<device\_id>/datapoints* |
| HTTP头部 | api-key:xxxx-ffff-zzzzz |
| HTTP内容 | **{**  **"datastreams":[**  **{**  **"id":"temperature",**  **"datapoints":[**  **{**  **"at":"2013-04-22T00:35:43",//参数可选**  **"value":42**  **},**  **{"at":"2013-04-22T00:55:43","value":84}**  **]**  **}，**  **{**  **"id":"key",**  **"datapoints":[**  **{"at":"2013-04-22T00:35:43","value":{…}},**  **{"at":"2013-04-22T00:35:43","value":{…}}**  **]**  **}，{…}**  **]**  **}** |
| 请求返回 | **{**  **"errno": 0,**  **"error":“succ”,**  **{}**  **}** |

### 更新

暂不支持。

### 查看

注：

设备云识别的GPS型数据格式（value）：

{“lon”:33.2,”lat”:23,”ele”:222}

二进制数据数据索引存储格式（value）：

{ “index”:”FW3SO72FZIF3W”,“size”:2333(字节), ”desc”:字符串或json对象(根据上传时的desc参数确定)}

|  |  |
| --- | --- |
| HTTP方法 | GET |
| URL | http://<API\_ADDRESS>/devices/<device\_id> /datapoints |
| URL参数 | ~~sort\_time = -1, // 可选，指定时按时间倒序排，最新时间在前面~~  ~~page = 1 //指定页码, 可选~~  ~~per\_page = 300 //指定每页输出数据点个数,可选, 默认300，最多1000~~  datastream\_id: <datastream\_id>,// 指定datastream,可选,多个用逗号分隔  start: *2012-06-02T14:01:46*,// 指定开始时间，必选，时间格式ISO 8601  end=*2012-06-02T14:01:46*,// 指定结束时间,可选，间隔最大6小时  duration=600,//时间间隔，可选，默认600，单位秒，最大6小时  limit=100,// datapoint个数，默认10，最大6000  cursor=XXXXXXX,//查询条件返回结果超过limit条数时，用于遍历的游标。  使用示例：  http://api.onenet.com/devices/<device\_id>/datapoints?datastream\_id=temperature&start=2013-05-12T17:22:33&duration=600 |
| HTTP头部 | api-key:xxxx-ffff-zzzzz |
| 请求返回 | **//该接口返回数据如下：**  **{**  **"errno": 0,**  **"error":“succ”,**  **"data":**  **{**  **"cursor":"XXAABBCCDD"，**  **~~"total\_count":3000,~~**  **~~"per\_page":300,~~**  **~~"page":1,~~**  **"datastreams":[**  **{**  **"id":"temperature",**  **"datapoints":[**  **{"at":" xxxx-xx-xx 10:22:22","value":42},**  **{"at":" xxxx-xx-xx 10:22:22","value":84}**  **]**  **}，**  **{**  **"id":"key",**  **"datapoints":[**  **{"at":" xxxx-xx-xx 10:22:22","value":{…}},**  **{"at":"xxxx-xx-xx 10:22:22","value":{…}}**  **]**  **}，{…}**  **]**  **}**  **}** |

### 删除

|  |  |
| --- | --- |
| HTTP方法 | DELETE |
| URL | http://<API\_ADDRESS>/ devices /<device\_id>/datapoints |
| URL参数 | //指定开始或结束时间，时间格式ISO 8601  datastream\_id: <datastream\_id>,// 指定删除datastream中的数据,可选  start: *2012-06-02T14:01:46*,// 指定开始时间  end: *2013-06-02T14:01:46*,// 指定开始时间  duration: 3600,// 时间长度，单位秒  start,end,duration参数组合方式：  start,end: 删除start开始到end结束的数据；  start,duration:删除start开始到start+duration结束的时间段数据；  end,duration:删除end-duration开始到end结束的时间段数据；  三个参数同时指定时，按组合方式1 处理  使用示例：  http://api.onenet.com/devices/<device\_id>/datapoints?  datastream\_id=temperature&start=2013-05-12T17:22:33&duration=60 |
| HTTP头部 | api-key:xxxx-ffff-zzzzz |
| HTTP内容 | **无** |
| 请求返回 | **{**  **"errno": 0,**  **"error":“succ”**  **}** |

## 触发器（trigger）

### 新增

|  |  |
| --- | --- |
| HTTP方法 | POST |
| URL | http://<API\_ADDRESS>/triggers |
| HTTP头部 | api-key:xxxx-ffff-zzzzz |
| HTTP内容 | **{**  **//三种监控方式，1:只有ds\_id，监控该用户下所有设备数据流名称为该id的数据；2:只有ds\_uuids,监控该用户下指定数据流uuid的数据； 3：ds\_id和dev\_ids，监控某些设备数据流名称为ds\_id的数据。**  **"ds\_id":"gps",//可选，数据流名称（id）**  **"dev\_ids":["1027","23456","15523"],//可选，设备ID**  **"ds\_uuids":[//可选，数据流uuid**  **"datastream1\_uuid”,**  **"datastream2\_uuid”,**  **"datastream3\_uuid”**  **],**  **"url":"http://xx.bb.com",**  **"type":">|>=|<|<=|==|inout|~~in|out~~|exp|change|frozen|live",**  **"threshold ":100** // 根据type不同，见以下说明。  **}**  注：   1. type为>|>=|<|<=|==时，threshold必须为数值 2. type为inout时，threshold设置为{“lolmt”:40, “uplmt”:52}，表示数据流的值首次进入或离开闭区间[40,52]时触发； 3. type为exp时,threshold设置为字符串类型的条件表达式，$val[0]表示第一个数据流的当前值，$val[1]为第二个…。 例如，第一个数据流上报数据点格式为：{“temperature”:22, “humidity”:56},若需设置温度大于30，且湿度小于33时触发告警，则threshold中条件表达式可设置为：“$val[0][‘temperature’] > 30 && $val[0][‘humidity’]< 33 ”； 4. ~~type为in|out时，需要设置中心和半径，threshold格式为：~~   ~~{~~  ~~"centre": {"lat":22.3121,"lon":3.244},~~  ~~"radius": 200,//单位（米）~~  ~~}~~   1. type为change时，threshold 参数不用传递；当上传的值有改变时触发告警。 2. type为frozen时，threshold 为数值，指定多少秒内未上报数据触发告警，同时被监控对象进入frozen状态。 3. type为live时，threshold不用传递；被监控对象在frozen状态下收到上报的数据点时，触发告警。 |
| 请求返回 | **{**  **"errno": 0,**  **"error":“succ”,**  **"data":**  **{**  **"trigger\_id":233444**  **}**  **}** |

### 更新

根据更新参数，全量设置对应项目。

|  |  |
| --- | --- |
| HTTP方法 | PUT |
| URL | http://<API\_ADDRESS>/triggers/<trigger\_id> |
| HTTP头部 | api-key:xxxx-ffff-zzzzz |
| HTTP内容 | **{**  **"ds\_uuids":[**  **"datastream1\_uuid” ,**  **"datastream2\_uuid” ,**  **"datastream3\_uuid”**  **],**  **"url":"http://xx.bb.com",**  **"type":">|>=|<|<=|==|in|out|exp|change|frozen|live ",**  **"threshold":100**  **}** |
| 请求返回 | **{**  **"errno": 0,**  **"error":“succ”**  **}** |

### 查看

|  |  |
| --- | --- |
| HTTP方法 | GET |
| URL | http://<API\_ADDRESS>/triggers/<trigger\_id> |
| HTTP头部 | api-key:xxxx-ffff-zzzzz |
| 请求返回 | **{**  **"errno": 0,**  **"error":“succ”,**  **"data":{**  **"id":trigger\_id1,**  **"ds\_uuids":[**  **"datastream1\_uuid” ,**  **"datastream2\_uuid” ,**  **"datastream3\_uuid”**  **],**  **"url":"xx.bb.com",**  **"type":">|>=|<|<=|==|in|out|exp|change|frozen|live ",**  **"threshold":100,**  **"invalid":true//触发器是否失效**  **}**  **}** |

### 删除

|  |  |
| --- | --- |
| HTTP方法 | DELETE |
| URL | http://<API\_ADDRESS>/triggers/<trigger\_id> |
| HTTP头部 | api-key:xxxx-ffff-zzzzz |
| HTTP内容 | **无** |
| 请求返回 | **{**  **"errno": 0,**  **"error":“succ”**  **}** |

## API 权限(key)

### 新增

|  |  |
| --- | --- |
| HTTP方法 | POST |
| URL | http://<API\_ADDRESS>/keys |
| HTTP头部 | api-key:xxxx-ffff-zzzzz  (注：必须是master-key) |
| HTTP内容 | **{**  **"title":"sharing key",**  **"permissions":[**  **{**  **"resources":[**  **{**  **"dev\_id":"424",**  **"ds\_id":"fan1"//可选**  **}**  **]**  **}**  **]**  **}** |
| 请求返回 | **{**  **"errno": 0,**  **"error":“succ”,**  **"data":**  **{**  **"key":"FJWEFOWFNWBSNOQ"**  **}**  **}** |

### 更新

根据更新参数，全量设置对应项目。

|  |  |
| --- | --- |
| HTTP方法 | PUT |
| URL | http://<API\_ADDRESS>/keys/<key\_string> |
| HTTP头部 | api-key:xxxx-ffff-zzzzz  （注：只能为master-key） |
| HTTP内容 | **{**  **"title":"sharing key",**  **"permissions":[**  **{**  **"resources":[**  **{**  **"dev\_id":"424",**  **"ds\_id":"fan1"//数据流id，可选**  **}**  **]**  **}**  **]**  **}** |
| 请求返回 | **{**  **"errno": 0,**  **"error":“succ”**  **}** |

### 查看

|  |  |
| --- | --- |
| HTTP方法 | GET |
| URL | http://<API\_ADDRESS>/keys |
| URL参数 | **dev\_id = xxbbcc,//可选,只查看与该设备相关的非master-key**  **key = FWJPO-FZFHW-FFEWB,//可选，只查看该key相关信息**  使用示例：  http://<API\_ADDRESS>/keys?dev\_id=10029 |
| HTTP头部 | api-key:xxxx-ffff-zzzzz  （注：只能为master-key） |
| 请求返回 | **{**  **"errno": 0,**  **"error":“succ” ,**  **"data":**  **[{**  **"key":"xxx—wwfw--wfwz",**  **"title":"sharing key",**  **"permissions":[**  **{**  **"resources":[//可选，没有，标示master-key**  **{**  **"dev\_id":"424",**  **"ds\_id":"fan1"**  **}**  **]**  **}**  **]**  **}]**  **}** |

### 删除

|  |  |
| --- | --- |
| HTTP方法 | DELETE |
| URL | http://<API\_ADDRESS>/keys/<key\_string> |
| HTTP头部 | api-key:xxxx-ffff-zzzzz  (注：只能为master-key) |
| HTTP内容 | **无** |
| 请求返回 | **{**  **"errno": 0,**  **"error":“succ”**  **}** |

## 二进制数据（bindata）

### 新增

功能：将二进制的数据保存到设备云，设备云返回该数据索引；并根据参数，将索引保关联到设备，或者作为数据点保存到设备下的某个数据流。

|  |  |
| --- | --- |
| HTTP方法 | POST |
| URL | http://<API\_ADDRESS>/bindata |
| HTTP头部 | api-key:xxxx-ffff-zzzzz |
| URL参数 | **device\_id = //可选，该数据所属设备。不指定data\_stream时，该数据索引被关联到设备的binary属性（见设备查看接口）。**  **datastream\_id = //可选，该数据所属数据流。必须与device\_id一起指定，该数据的索引被作为datapoint保存在数据流中（数据点格式见查看数据点）。**  **//若没有指定device\_id和datastream\_id，该数据的索引被保存到用户级别（见用户查看接口）。**  **desc = //可选，描述字符串或json字符串。以字符串或解析后的json对象格式，与生成的index一起保存在设备属性或者数据点中。**  **at = //可选，时间，如“2014-10-23 10:22:23”。默认当前时间** |
| HTTP内容 | 普通二进制数据、文件、图像 |
| 请求返回 | **{**  **"errno": 0,**  **"error":“succ”,**  **"data":{**  **"index":“AJOWFW1133OFO2Z93”**//该数据在设备云的索引  **}**  **}** |

### 查看

功能：读取二进制数据的内容。若查询没有该数据返回HTTP 404,并在返回内容中指定{errno,error}

|  |  |
| --- | --- |
| HTTP方法 | GET |
| URL | http://<API\_ADDRESS>/bindata/<index> |
| HTTP头部 | api-key:xxxx-ffff-zzzzz |
| URL参数 |  |
| HTTP内容 |  |
| 请求返回 | 普通二进制数据、文件、图像 |

### 删除

功能说明：目前只用于删除用户或设备级别的二进制数据

|  |  |
| --- | --- |
| HTTP方法 | DELETE |
| URL | http://<API\_ADDRESS>/bindata/<index> |
| HTTP头部 | api-key:xxxx-ffff-zzzzz  (注：只能为master-key) |
| HTTP内容 | **无** |
| 请求返回 | **{**  **"errno": 0,**  **"error":“succ”**  **}** |

## 其他

### 传感器触发（异常事件内容）

**{**

**"trigger":{**

**"id":1,**

**"threshold":9.0,**

**"type":">"**

**},**

**"current\_data":[**

**{**

**"dev\_id":"2656",**

**"ds\_id":"gps",**

**"ds\_uuid":"FWEF-ZFW-ZFW-ZFWW",**

**"at":"2009-09-07T12:16:02.001403Z",**

**"value":100**

**},**

**…**

**{**

**"dev\_id":"2656",**

**"ds\_id":"temper",**

**"ds\_uuid":"FEWEF-ZFWFWW-WFWW-FWEFFWW",**

**"at":"2009-09-07T12:16:02.001403Z",**

**"value":88.2**

**}**

**}**

### 发送数据到EDP设备

|  |  |
| --- | --- |
| HTTP方法 | POST |
| URL | http://<API\_ADDRESS>/cmds/<device\_id> |
| HTTP头部 | api-key:xxxx-ffff-zzzzz |
| HTTP内容 | 用户自定义Json或二进制数据 |
| 请求返回 | **{**  **"errno": 0,**  **"error":“succ”**  **}** |

### RestAPI 日志

功能和限制说明：保存某个设备操作相关的restAPI日志，目前只保存最近半个小时日志，每次查询最多输出时间参数最近20条。

|  |  |
| --- | --- |
| HTTP方法 | GET |
| URL | http://<API\_ADDRESS>/logs/<device\_id> |
| HTTP头部 | api-key:xxxx-ffff-zzzzz |
| ~~URL参数~~ | **t\_start = 2014-12-12 20:22:22 //可选，查询从某个时间开始，最近的20条日志**  **~~没有参数指定时，默认输出最近20条~~** |
| HTTP内容 | null |
| 请求返回 | **{**  **"errno": 0,**  **"error":“succ”**  **"**data**":**[//输出最近该设备restful操作日志20条，按时间倒排  [{"log\_time":"2014-08-29 12:49:32",  "url":"\/transfer\/25267",  "method":"POST",  "request\_time":"2014-08-29 12:49:32",  "request\_header":["User-Agent: curl\/7.19.7 (x86\_64-redhat-linux-gnu) libcurl\/7.35.0 OpenSSL\/1.0.0 zlib\/1.2.3 libidn\/1.18","Host: 10.189.24.66","Accept: \*\/\*",  "api-key: Bs04OCJioNgpmvjRphRak15j7Z8="],  "request\_body":"",  "return\_code":"200",  "response\_time":"2014-08-29 12:49:32",  "response\_header":["X-Powered-By: PHP\/5.5.9"],  "response\_body":"{\"errno\":1,\"error\":\"no url with the device\"}"}],  {...}  ]  **}** |
|  |  |

### 历史数据查询

说明：根据条件查询该用户下的数据点记录。

|  |  |
| --- | --- |
| HTTP方法 | GET |
| URL | http://<API\_ADDRESS> /datapoints |
| URL参数 | ~~page = 1 //指定页码, 可选~~  device\_id= <device\_id>,// 指定设备ID,可选  datastream\_id= <datastream\_id>,// 指定数据流,可选  start=*2012-06-02T14:01:46*,// 指定开始时间,必选  end=*2012-06-02T14:01:46*,// 指定结束时间,可选，间隔最大6小时  duration=600,//时间间隔，可选，默认600，单位秒，最大6小时  limit=100,// datapoint个数，默认10，最大6000  cursor=XXXXXXX,//查询条件返回结果超过limit条数时，用于遍历的游标。  使用示例，查询该用户下，所有设备gps数据流中的数据点：  http://api.onenet.com /datapoints?datastream\_id=gps |
| HTTP头部 | api-key:xxxx-ffff-zzzzz （必须使用master\_key） |
| 请求返回 | **//示例查询返回数据如下：**  **{**  **"errno": 0,**  **"error":“succ”,**  **"data":**  **{**  **"cursor":"XXAABBCCDD"，**  **"datapoints":[**  **{**  **"dev\_id":"10313",//设备ID**  **"ds\_id":"gps",//数据流ID**  **"at":" xxxx-xx-xx 10:22:22",//数据点时间**  **"value":{"lat":23.1,"lon":89.2","ele":200}，**  **},{**  **"dev\_id":"2656",**  **"ds\_id":"gps",**  **"at":" xxxx-xx-xx 10:22:22",**  **"value":{"lat":23.1,"lon":69.2","ele":120}，**  **},{…}**  **]**  **}**  **}** |