

# Diacloud API 接口说明文档

## V1.4.1

修改版本	修改日期	修改内容
V1.3	2020-06-05	<ul style="list-style-type: none"><li>➢ 增加章节编号</li><li>➢ 新增:<ul style="list-style-type: none"><li>➢ F4: Dword 和浮点型的算法</li><li>➢ F5: 设备寄存器当前数据接口, 获取不到 M 区数据?</li><li>➢ F6: 设备寄存器当前数据接口, 获取数据不全?</li><li>➢ F7: API 调用限制</li></ul></li></ul>
V1.4	2020-06-17	<ul style="list-style-type: none"><li>➢ 接口的时区声明</li><li>➢ 接口更名: 只做命名修改, 其它不变<ul style="list-style-type: none"><li>➢ 原接口“获取寄存器历史数据接口”更名为“获取单个历史数据接口”</li><li>➢ 原接口“获取设备寄存器当前数据接口”更名为“按 UTC+8 时钟获取设备寄存器当前数据接口”</li><li>➢ 原接口“全部设备 alarm 信息接口”更名为“按 UTC+8 时钟获取全部设备 alarm 信息接口”</li><li>➢ 原接口“单台设备 alarm 信息接口”更名为“按 UTC+8 时钟获取单台设备 alarm 信息接口”</li></ul></li><li>➢ 接口变更:<ul style="list-style-type: none"><li>➢ 获取单个历史数据接口, 新增支持固定时间间隔查询参数</li></ul></li><li>➢ 接口新增<ul style="list-style-type: none"><li>➢ 新增 1: 按寄存器配置获取寄存器历史</li><li>➢ 新增 2: 按 UTC 时钟获取设备寄存器当前数据接口</li><li>➢ 新增 3: 按 UTC 时钟获取全部设备 alarm 信息接口</li><li>➢ 新增 4: 按 UTC 时钟获取单台设备 alarm 信息接口</li><li>➢ FAQ-F8: 如何上传通讯错误标志</li></ul></li></ul>
V1.4.1	2020-0728	<ul style="list-style-type: none"><li>➢ 修正 DIACloud API 接口说明中 UTC+0 指令说明</li></ul>

## 目录

一、DIACloud API 接口说明.....	3
1.1 Diaclou API 接口简介.....	3
1.2 设备基本信息接口.....	4
1.3 按 UTC+8 获取全部设备 alarm 信息接口.....	6
1.4 按 UTC+8 获取单台设备 alarm 信息接口.....	9
1.5 安全隧道列表信息接口.....	12
1.6 按 UTC+8 获取设备寄存器当前数据接口.....	13
1.7 设置设备寄存器数值接口.....	16
1.8 获取寄存器历史数据接口.....	17
1.9 删除设备某个寄存器接口.....	20
1.10 获取每个设备的某几个寄存器的值.....	20
1.11 按 UTC 获取全部设备 alarm 信息接口.....	22
1.12 按 UTC 获取单台设备 alarm 信息接口.....	25
1.13 按 UTC 获取设备寄存器当前数据接口.....	28
1.14 按寄存器配置获取历史数据接口.....	30
1.14.1 寄存器的配置步骤.....	30
1.14.2 按寄存器配置获取历史数据接口介绍.....	31
FAQ: 常见问题.....	34
F1: 调用 API 提示“please replace the variable in {} with the correct value”.....	34
F2: 调用 API 返回 403fobind.....	35
F3: 显示设置成功, 但没有生效.....	35
F4: Dword 和浮点型的算法.....	35
F5: 设备寄存器当前数据接口, 获取不到 M 区数据?.....	36
F6: 设备寄存器当前数据接口, 获取数据不全?.....	36
F7: API 调用限制.....	37
F8: 如何上传通讯错误标志.....	37

# 一、DIACloud API 接口说明

## 1.1 Diaclou API 接口简介

API 接口说明地址: <https://api.diacloudsolutions.com.cn/>

API 接口分类: 设备基本信息接口、全部设备 alarm 信息接口、单台设备 alarm 信息接口、安全隧道列表信息接口、获取设备寄存器当前数据接口、设置设备寄存器数值接口、获取寄存器历史数据接口、删除设备接口。介绍如下表:

名称	Restful Api	Json 请求类型	说明
设备基本信息接口	/devices	get	获取该账号下的设备的基本信息
按 UTC+8 获取全部设备 alarm 信息接口	/alarms	get	按 UTC+8 获取该账号下全部 alarms
按 UTC+8 获取单台设备 alarm 信息接口	/devices/{device_id}/alarms	get	按 UTC+8 获取该账号下指定设备的 alarms
安全隧道列表信息接口	/vlans	get	获取该账号下所有安全隧道的信息
按 UTC+8 时钟获取设备寄存器当前数据接口	/devices/{device_id}/regs	get	按 UTC+8 时钟获取该账号下某台设备寄存器的当前值。
设置设备寄存器数值接口	/devices/{device_id}/regs	put	设备该账号下某台设备某个寄存器的值
获取单个历史数据接口	/devices/{device_id}/regs/{reg_addr}/history	get	按变化时间获取该账号下某台设备某个寄存器的历史数据。(时钟为 UTC+0)
删除设备寄存器接口	/devices/{device_id}/regs/{reg_addr}	delete	删除该账号下某个设备某个寄存器的数据
按寄存器配置获取寄存器历史	/devices/{device_id}/regshistory	get	按固定的时间间隔返回某一账号, 某一设备固定的间隔时间的多个寄存器历史数据。(时钟为 UTC+0)
按 UTC 时钟获取设备寄存器当前数据接口	/v2/devices/{device_id}/regs	get	按 UTC+0 时钟获取该账号下某台设备寄存器的当前值。
按 UTC 时钟获取全部设备 alarm 信息接口	/v2/alarms	get	按 UTC+0 时钟获取该账号下全部 alarms

按 UTC 时钟获取 单台设备 alarm 信息接口	/v2/devices/{device_id}/alarms	get	按 UTC+0 时钟获取该账号下 指定设备的 alarms
----------------------------------	--------------------------------	-----	----------------------------------

## 1.2 设备基本信息接口

用于获取该账号下的设备的基本信息

Json 请求格式:

url: <https://api.diacloudsolutions.com.cn/devices>

方法: get

请求头:

request headers	
Accept	application/json
Content-Type	application/json
Authorization	Basic ZGIhY2xvdWRAMTYzLmNvbTp0ZXN0MTIzNDU2

注: authoriz 为认证, 后面是 ( “用户名” : ” 密码” ) 的 base64 加密结果。

云端 api 接口的使用简介:

打开 url: <https://api.diacloudsolutions.com.cn/>

选择 api list 中的第一个 **GET** /devices, 选择右侧的 “basic auth”



输入在 diacloud 上注册的云账号, 点击【Refresh headers】

Instructions Basic Auth

diacloud@163.com

\*\*\*\*\*

Refresh headers

https://api.diacloudsolutions.com/devices GET send Headers URL params Post data

Takes: ...ms

Result: ...

将会生成 request headers,

Instructions Basic Auth

diacloud@163.com

\*\*\*\*\*

Refresh headers

https://api.diacloudsolutions.com/devices GET send Headers URL params Post data

request headers

Accept	application/json
Content-Type	application/json
Authorization	Basic ZGlnY2xvdWRAMTYzLmNvbTp0ZlN0MTZlN0U2

Takes: ...ms

Result: ...

点击【Send】后，将会显示返回的 json:

Takes: 232 ms

Result:

```
{
  "data": [
    {
      "id": 21669,
      "name": "厦门",
      "sn": "00002010f0f0050",
      "tunnel_id": 0,
      "storage_state": 1,
      "traffic_state": 1,
      "online": 0,
      "ip": "0.0.0.0"
    },
    {
      "id": 21907,
      "name": "厦门市湖里区污水厂",
      "sn": "00002010t320027",
      "tunnel_id": 1446,
      "storage_state": 1,
      "traffic_state": 1,
      "online": 0,
      "ip": "102.168.199.188"
    },
    {
      "id": 21940,
      "name": "集美污水厂",
      "sn": "00002010c00006",
      "tunnel_id": 2573,
      "storage_state": 1,
    }
  ]
}
```

其中 data 涉及到字段说明如下：

data 中条目	名称	说明
id	设备 id	设备在云平台的唯一标识
name	设备名	设备的别名
sn	设备 SN	设备硬件的唯一标识
tunnel id	安全隧道 ID	设备所在安全隧道的 ID 号
storage_state	云存储空间状态	设备所用云存储空间是否满 1:正常, 0: 用完
traffic_state	云流量状态	设备所用云流量状态 1:正常, 0: 用完
online	在线状态	设备是否在线: 0 离线, 1 在线
IP	设备 IP	设备的 IP 地址
eip	公网入口 IP	注: 只有最新版本 V1.4 以上版本才有该信息 比如: TCP:211. 97. 129. 248, 设备使用 TCP 方式连接到服务器, 其对应出口的 IP 是 211. 97. 129. 248
latency	时延	注: 只有最新版本 V1.4 以上版本才有该信息 设备连接到服务器的时延, 单位 ms

### 1.3 按 UTC+8 获取全部设备 alarm 信息接口

用于获取该账号下的设备的全部的 alarms 信息。

注意：该 Alarm 指的是在设备网页中设置的告警。详设置在设备网页中，【系统设置】->【事件管理】，事件类型中选择“告警事件”。



Json 请求格式：

url: <https://api.diacloudsolutions.com.cn/alarms>

方法: get

请求头:

## request headers

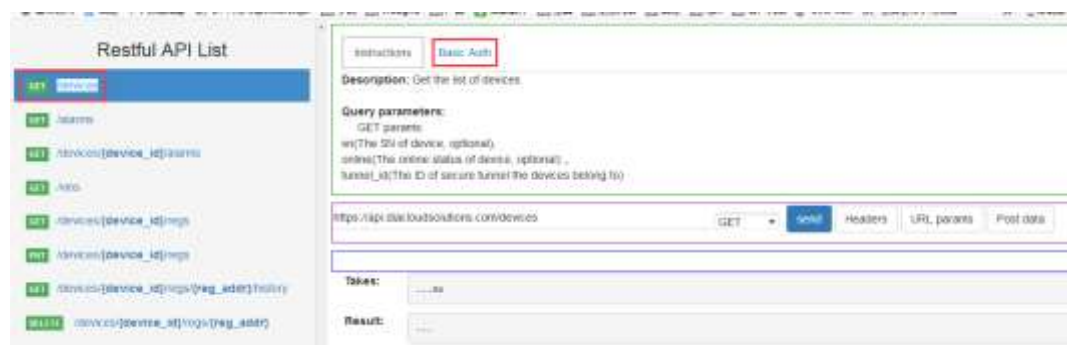
Accept	application/json
Content-Type	application/json
Authorization	Basic ZGIhY2xvdWRAMTYzLmNvbTp0ZXN0MTIzNDU2

注：authoriz 为认证，后面是（“用户名”：“密码”）的 base64 加密结果。

## 云端 api 接口的使用简介：

打开 url: <https://api.diacloudsolutions.com.cn/>

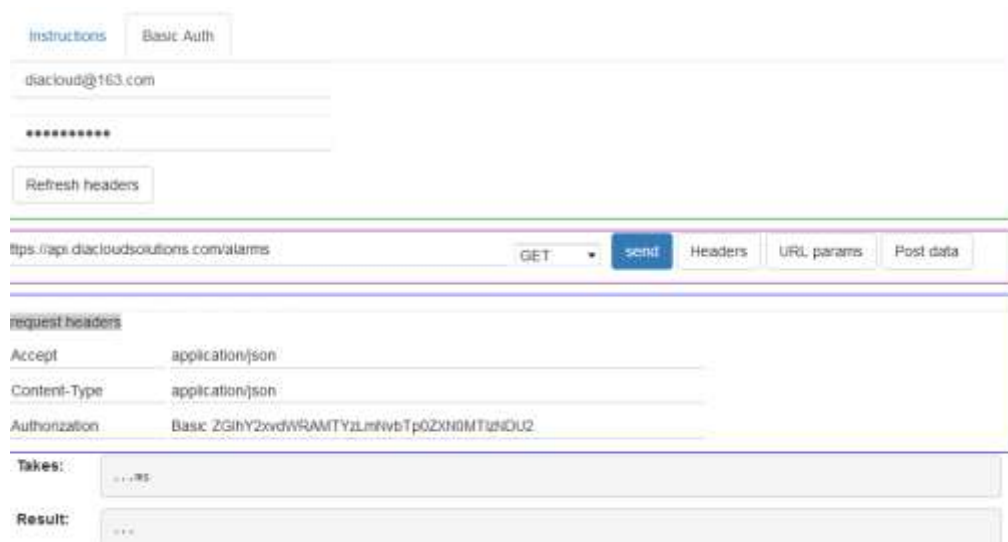
选择 api list 中的第二个 **GET /alarms** ,选择右侧的“basic auth”



输入在 diacloud 上注册的云账号，点击【Refresh headers】



将会生成 request headers,



点击【Send】后，将会显示返回的 json：



其中返回请求包含两部分

### 1、Paging 分页信息

paging 中条目	名称	说明
page	第几页	
limt	每页多少条	
total	一共有多少条告警	
pagecount	一共有多少页	
offset	偏移	

### 2、Data 中的含义

data 中条目	名称	说明
device_id	设备 id	发生告警的设备的 ID
name	设备名	短信告警的名称



title	告警标题	告警标题
content	告警内容	告警内容
sms.result	短信发送状态	短信是否发送成功 0:成功 1 是失败（使用云短信时才有记录）
sms.addr	短信发送的手机号	短信发送的目的手机号
email.result	邮件发送结果	邮件是否发送成功 0:成功 1 是失败（使用云短信时才有记录）
email.addr	邮件发送地址	邮件发送的目的邮箱
created	发送时间	告警发送的时间。该时间的格式为 UTC/GMT+08:00（东八区时间）

注：告警支持分页、按时间查询等方式

如果查询第二页的话则

<https://api.diacloudsolutions.com.cn/alarms?page=2>

如果查询 2018-01-01 到 2018-01-08

<https://api.diacloudsolutions.com.cn/alarms?from=2018-01-01 00:00:00 & to=2018-01-09 00:00:00>

多个条件之间用&连接

## 1.4 按 UTC+8 获取单台设备 alarm 信息接口

用于获取该账号下的某台设备的全部的 alarms 信息。

注意：告警来源和 [1.3 按 UTC+8 获取全部设备 alarm 信息接口](#) 中的告警均来源设备网页中的设置的告警。具体设置参见 [1.3 按 UTC+8 获取全部设备 alarm 信息接口](#)。

Json 请求格式：

url: [https://api.diacloudsolutions.com.cn/devices/{device\\_id}/alarms](https://api.diacloudsolutions.com.cn/devices/{device_id}/alarms)

方法: get

请求头:

request headers	
Accept	application/json
Content-Type	application/json
Authorization	Basic ZGIhY2xvdWRAMTYzLmNvbTp0ZXN0MTIzNDU2

注: authoriz 为认证, 后面是 ( “用户名” : ”密码” ) 的 base64 加密结果。

注: 其中 {device\_id} 请更换为设备信息接口中 data 中的 id。比如 data 中的 ID 为 21669

```
Takes: 232 ms
Result:
{
  "data": [
    {
      "id": 21669,
      "name": "厦门",
      "sn": "DXR02010F0F0030",
      "tunnel_id": 0,
      "storage_state": 1,
      "traffic_state": 1,
      "online": 0,
      "ip": "0.0.0.0"
    },
    {
      "id": 21907,
      "name": "厦门市湖里区污水厂",
      "sn": "DXR02010E320027",
      "tunnel_id": 1446,
      "storage_state": 1,
      "traffic_state": 1,
      "online": 0,
      "ip": "192.168.199.180"
    },
    {
      "id": 21940,
      "name": "曲靖污水厂",
      "sn": "DXR02010E080006",
      "tunnel_id": 2373,

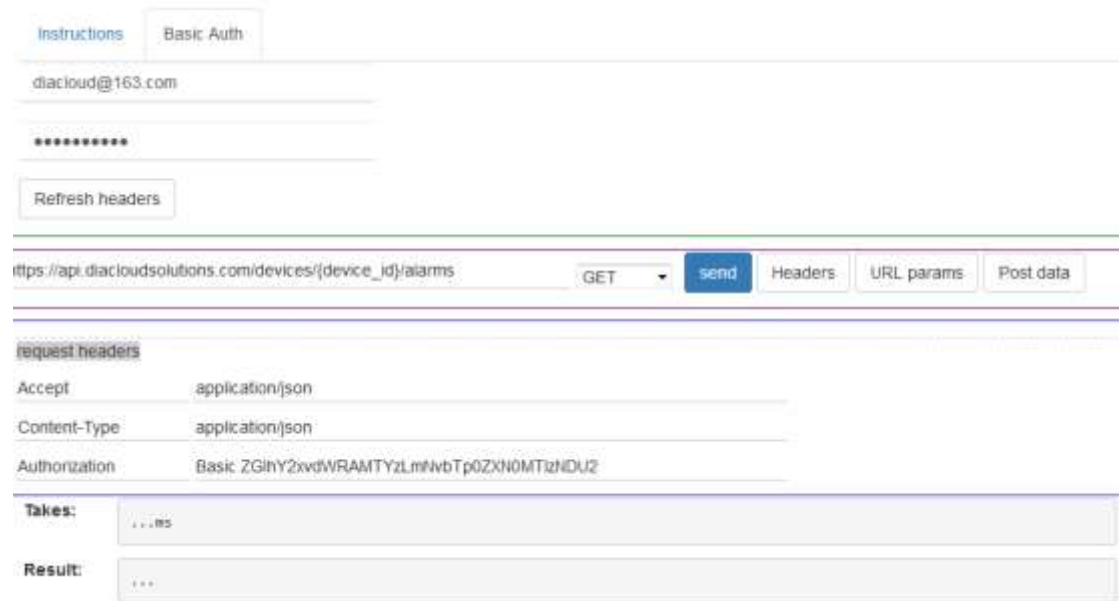
```

则 url 为 <https://api.diacloudsolutions.com.cn/devices/21669/alarms>

### 云端 api 接口的使用简介:

打开 url: <https://api.diacloudsolutions.com.cn/>

选择 api list 中的第三个 `GET /devices/{device_id}/alarms`, 选择右侧的“basic auth”, 并输入账号



在 url 中更换设备的 ID 号 21907 (有告警的设备 ID, 设备 ID 的查询方法请参看第一个基本信息接口)



点击【Send】后，将会显示返回的 json:

```

103 ms

{
  "paging": {
    "page": 1,
    "limit": 20,
    "total": "38",
    "pageCount": 2,
    "offset": 0
  },
  "data": [
    {
      "device_id": "21907",
      "name": "online",
      "title": "online",
      "content": "online 2017/01/04 17:10:22",
      "sms": {
        "result": 0,
        "addr": "13616052915"
      },
      "email": {
        "result": 0,
        "addr": "TO:{1160260354@qq.com;}"
      },
      "created": "2017-01-04 17:10:56.410524"
    },
    {
      "device_id": "21907",
      "name": "online",
      "title": "online",
  
```

其中返回请求包含两部分

### 3、Paging 分页信息

paging 中条目	名称	说明
page	第几页	
limt	每页多少条	
total	一共有多少条告警	
pagecount	一共有多少页	
offset	偏移	

### 4、Data 中的含义

data 中条目	名称	说明
device_id	设备 id	发生告警的设备的 ID

name	设备名	短信告警的名称
title	告警标题	告警标题
content	告警内容	告警内容
sms.result	短信发送状态	短信是否发送成功 0:成功 1 是失败（使用云短信时才有记录）
sms.addr	短信发送的手机号	短信发送的目的手机号
email.result	邮件发送结果	邮件是否发送成功 0:成功 1 是失败（使用云短信时才有记录）
email.addr	邮件发送地址	邮件发送的目的邮箱
created	发送时间	告警发送的时间。该时间的格式为 UTC/GMT+08:00（东八区时间）

注：告警支持分页、按时间查询等方式

如果查询第二页的话则

<https://api.diacloudsolutions.com.cn/devices/21097/alarms?page=2>

如果查询 2018-01-01 到 2018-01-08

<https://api.diacloudsolutions.com.cn/devices/21097/alarms?from=2018-01-01 00:00:00 &to=2018-01-09 00:00>

多个条件之间用&连接

## 1.5 安全隧道列表信息接口

用于获取该账号下的安全隧道的基本信息

Json 请求格式：

url: <https://api.diacloudsolutions.com.cn/vlms>

方法: get

请求头:

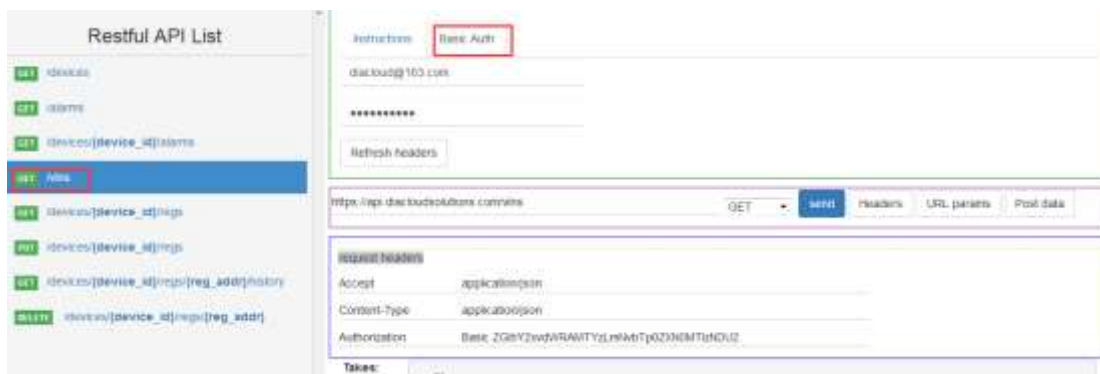
request headers	
Accept	application/json
Content-Type	application/json
Authorization	Basic ZGIhY2xvdWRAMTYzLmNvbTp0ZXN0MTIzNDU2

注: authoriz 为认证, 后面是 (“用户名” :” 密码”) 的 base64 加密结果。

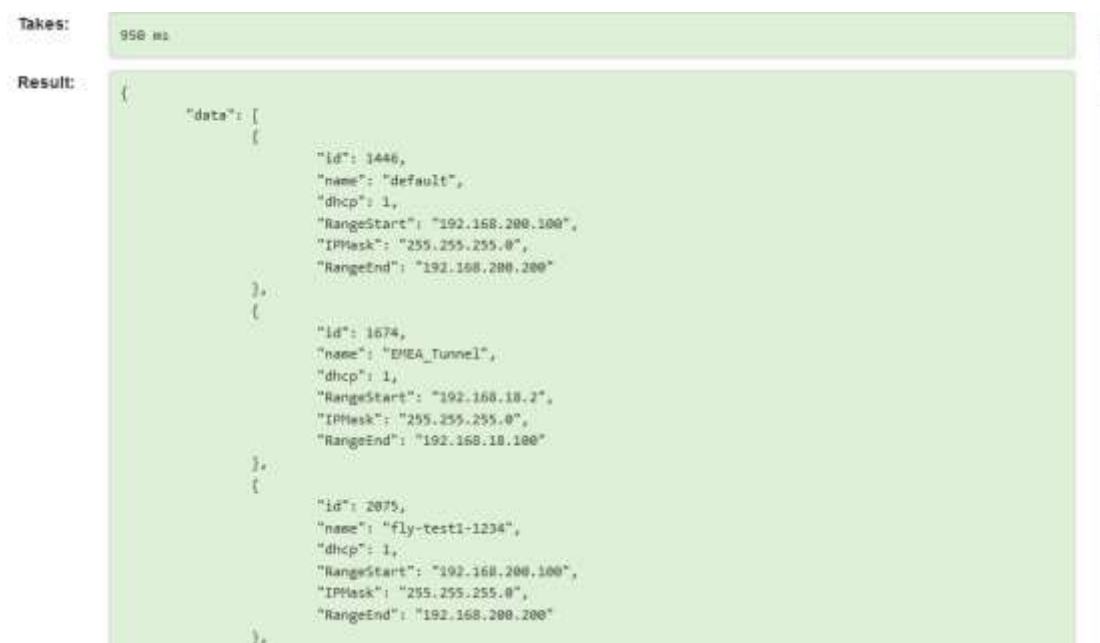
云端 api 接口的使用简介:

打开 url: <https://api.diacloudsolutions.com.cn/>

选择 api list 中的第四个   , 选择右侧的 “basic auth”



填写完账号和密码后，点击【Send】后，将会显示返回的 json:



其中 data 涉及到字段说明如下:

data 中条目	名称	说明
id	安全隧道 ID	安全隧道 ID 号
name	安全隧道名	安全隧道名
dhcp	DHCP 是否开启	1: DHCP 开启, 0: DHCP 停止
RangeStart	开始 IP	DHCP 的开始的 IP 地址
IPMask	子网掩码	子网掩码
RangeEnd	结束 IP	DHCP 的结束的 IP 地址

## 1.6 按 UTC+8 获取设备寄存器当前数据接口

用于获取该账号下的特定设备当前寄存器信息

Json 请求格式:

url: [https://api.diacloudsolutions.com.cn/devices/{device\\_id}/regs](https://api.diacloudsolutions.com.cn/devices/{device_id}/regs)

方法: get

请求头:

request headers	
Accept	application/json
Content-Type	application/json
Authorization	Basic ZGIhY2xvdWRAMTYzLmNvbTp0ZXN0MTIzNDU2

注: authoriz 为认证, 后面是 (“用户名”:”密码”) 的 base64 加密结果。

注: 其中 {device\_id} 请更换为设备信息接口中 data 中的 id。比如 data 中的 ID 为 21669

Takes: 232 ms

Result:

```
{
  "data": [
    {
      "id": 21669,
      "name": "厦门",
      "sn": "DXR02010F0F0030",
      "tunnel_id": 0,
      "storage_state": 1,
      "traffic_state": 1,
      "online": 0,
      "ip": "0.0.0.0"
    },
    {
      "id": 21907,
      "name": "厦门市湖里区污水厂",
      "sn": "DXR02010E320027",
      "tunnel_id": 1446,
      "storage_state": 1,
      "traffic_state": 1,
      "online": 0,
      "ip": "192.168.199.100"
    },
    {
      "id": 21940,
      "name": "曲靖污水厂",
      "sn": "DXR02010E000006",
      "tunnel_id": 2373,

```

则 url 为 <https://api.diacloudsolutions.com.cn/devices/21669/regs>

注 1: 默认返回数据从 2048 开始, 如果需要读取 bit 区域数据 (\$1536~\$2047) 需要添加 start 参数, 如:

<https://api.diacloudsolutions.com.cn/devices/21669/regs?start=1536>.

注 2: 如果是账号错误或者设备 ID 错误将返回 403 error。

注 3: 该接口不可以用于获取 \$1536 之前系统寄存器的数据。

### 云端 api 接口的使用简介:

打开 url: <https://api.diacloudsolutions.com.cn/>

选择 api list 中的第五个 **GET** `/devices/{device_id}/regs`, 选择右侧的“basic auth”

填写完账号和密码后，填入设备 ID 后，点击【Send】后，将会显示返回的 json:

```

Takes: 466 ms
Result:
{
  "count": 28,
  "data": [
    {
      "addr": 2048,
      "value": 2017,
      "time": "2017-01-10 08:43:26",
      "name": null,
      "template": null,
      "history": 1
    },
    {
      "addr": 2049,
      "value": 1,
      "time": "2017-01-10 08:43:26",
      "name": "$2049",
      "template": "return val;",
      "history": 1
    },
    {
      "addr": 2050,
      "value": 10,
      "time": "2017-01-10 08:43:26",
      "name": null,
      "template": null,
      "history": 1
    }
  ]
}

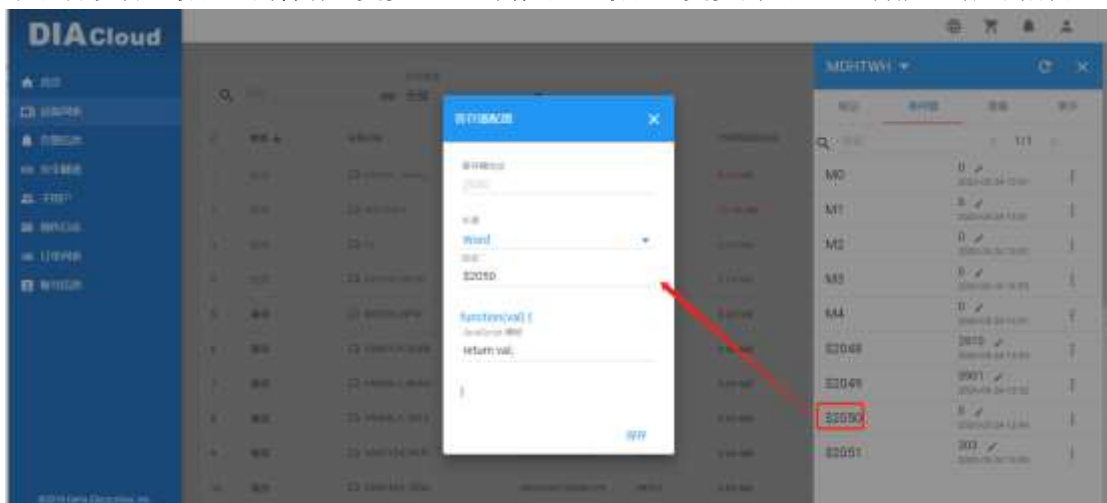
```

响应分成两个部分，count 和 data，中 count 是寄存器的个数。

其中 data 涉及到字段说明如下：

data 中条目	名称	说明
Addr	寄存器地址	寄存器地址 (2048~4096)
Value	寄存器值	寄存器值 (无符号十进制整数，其他类型需要自己转换)
Time	时间	寄存器数值上传的时间。该时间的格式为 UTC/GMT+08:00 (东八区时间)。
templatet	转换脚本	原始值需要加工转后显示的脚本
History	是否保存历史	1、保存历史，0：不保存历史

其中 name 的属性需要在 <https://api.diacloudsolutions.com.cn> 的网站上【设备列表】中，选中该台设备，最右边的操作【更多】->【寄存器】，最右边更多的【配置】功能，对应的别名。



## 1.7 设置设备寄存器数值接口

用于设置该账号下特定的设备和寄存器信息

Json 请求格式:

url: [https://api.diacloudsolutions.com.cn/devices/{device\\_id}/regs](https://api.diacloudsolutions.com.cn/devices/{device_id}/regs)

方法: put

请求头:

request headers	
Accept	application/json
Content-Type	application/json
Authorization	Basic ZGIhY2xvdWRAMTYzLmNvbTp0ZXN0MTIzNDU2

Body 中填写需要发送的数据

```
[{"addr":2048,"value":100},{"addr":2049,"value":100}]
```

post data(Please input valid JSON object)

```
[{"addr":2048,"value":100},{"addr":2049,"value":100}]
```

注: authoriz 为认证, 后面是 ( “用户名” : ” 密码” ) 的 base64 加密结果。

注: 其中 {device\_id} 请更换为设备信息接口中 data 中的 id。比如 data 中的 ID 为 21669

则 url 为 <https://api.diacloudsolutions.com.cn/devices/21669/regs>

云端 api 接口的使用简介:

打开 url: <https://api.diacloudsolutions.com.cn/>

选择 api list 中的第六个 **PUT** `/devices/{device_id}/regs`, 选择右侧的 “basic auth”



Instructions
Basic Auth

---

https://api.diacloudsolutions.com/devices/27709/regs
PUT
send
He

**request headers**

Accept	application/json
Content-Type	application/json
Authorization	Basic ZGIhY2xvdWRAMTYzLmNvbTp0ZXN0MTIzNDU2

**post data**(Please input valid JSON object)

```
[{"addr":2048,"value":100},{"addr":2049,"value":100}]
```

填写完账号和密码后，填入设备 ID 后，还有 postdata 中的内容。点击【Send】后，将会返回结果。202 表示操作完成，如果是其它设置失败，请确认网络、设置的参数是否正确。

**Takes:** 300 ms

**Result:** 202 Accepted

## 1.8 获取寄存器历史数据接口

用于设置该账号下特定的设备的某个寄存器历史信息

Json 请求格式:

url:[https://api.diacloudsolutions.com.cn/devices/{device\\_id}/regs/{reg\\_addr}/history](https://api.diacloudsolutions.com.cn/devices/{device_id}/regs/{reg_addr}/history)

请求参数:

page: 可选

limit: 每页返回数量, 可选

from: 查询历史的开始时间参数, 可选。时间格式为 UTC 时钟, 比如: 2019-01-23 05:40:48。

to: 查询历史的开始时间参数, 可选。时间格式为 UTC 时钟, 比如: 2019-01-23 05:40:48。

gap: 固定的时间间隔, 单位为秒。如果该参数缺省, 默认返回变化的时间戳; 如果该参数填了后, 数据就会按设定的间隔返回数据。

方法: get

请求头:

request headers	
Accept	application/json
Content-Type	application/json
Authorization	Basic ZGhY2xvdWRAMTYzLmNvbTp0ZXN0MTIzNDU2

注: authoriz 为认证, 后面是 ( “用户名” :” 密码” ) 的 base64 加密结果。

注: 其中 {device\_id} 请更换为设备信息接口中 data 中的 id。比如 data 中的 ID 为 21669

注: 其中 {reg\_addr} 寄存器的地址

假设需要获取设备 ID 为 21907 的 2048 寄存器的历史数据则 url 为

<https://api.diacloudsolutions.com.cn/devices/21907/regs/2048/history>

云端 api 接口的使用简介:

打开 url: <https://api.diacloudsolutions.com.cn/>

选择 api list 中的第六个 **PUT** `/devices/{device_id}/regs`, 选择右侧的“basic auth”



填写完账号和密码后, 填入设备 ID 后和寄存器编号, 点击【Send】后, 将会显示返回的 json:

```

{
  "paging": {
    "page": 1,
    "limit": 1000,
    "total": 50,
    "pageCount": 1,
    "offset": 0
  },
  "data": [
    [
      "2017-01-10T00:43:26.819317Z",
      "21907",
      "2048",
      2017
    ],
    [
      "2017-01-10T00:43:23.872064Z",
      "21907",
      "2048",
      0
    ],
    [
      "2017-01-10T00:21:45.027323Z",
      "21907",
      "2048",
      2017
    ],
    [
      "2017-01-10T00:21:39.679019Z",
      "21907",
      "2048",
      0
    ]
  ]
}

```

响应分成两个部分， paging 和 data， 中 count 是寄存器的个数。

#### 1、Paging 分页信息

paging 中条目	名称	说明
page	第几页	
limit	每页多少条	
total	一共有多少条数据	
pagecount	一共有多少页	
offset	偏移	

上面图中的 data 涉及到字段说明如下：

data 中条目	值	说明
时间	2017-01-10T00:43:26.819317Z	寄存器上传的时间，含义为 2017 年，1 月 10 日 00:43:26 时区为 0 时区
设备 ID	21907	寄存器数值上传的时间
寄存器编号	2048	寄存器的编号为 2048
寄存器值	2017	寄存器的值为 2017

注：寄存器支持分页、按时间查询等方式

如果查询第八页的话则

<https://api.diacloudsolutions.com.cn/devices/21907/regs/2048/history?page=8>

如果查询 2018-01-01 到 2018-01-08

<https://api.diacloudsolutions.com.cn/devices/21907/regs/2048/history?from=2018-01-01>

00:00:00 & to=2018-01-09 00:00

如果 from 和 to 的参数均缺省的情况，默认返回最近一小时内的数据，数据最多 2000 条

如果按等时间间隔，的请求参数如下。查询 2020 年 2020-06-16 一天的数据，固定间隔为 60 秒。

<https://api.diacloudsolutions.com.cn/devices/21907/regs/2048/history?from=2020-06-16 00:00:00 & to=2020-06-17 00:00:00&gap=60>

ps: 单次调用历史数据，返回条数最多 1 万条。

## 1.9 删除设备某个寄存器接口

该接口目前尚未开放

## 1.10 获取每个设备的某几个寄存器的值

该接口 api 网站上还未公布。

用于和获取该账号下所有设备的特定寄存器的值。

Json 请求格式:

URL: [https://api.diacloudsolutions.com.cn/regs?start={reg\\_start}&end={reg\\_end}](https://api.diacloudsolutions.com.cn/regs?start={reg_start}&end={reg_end})

方法: get

请求头:

request headers	
Accept	application/json
Content-Type	application/json
Authorization	Basic ZGIhY2xvdWRAMTYzLmNvbTp0ZXN0MTIzNDU2

注: authoriz 为认证，后面是（“用户名” : ”密码”）的 base64 加密结果。

注: 其中 {reg\_start} 需要获取寄存器编号的开始地址

注: 其中 {reg\_end} 需要获取寄存器编号的结束地址

假设需要获取自己账号下的所有设备设备 ID 为 2048 和 2049 寄存器的历史数据则 url 为

<https://api.diacloudsolutions.com.cn/regs?start=2048&end=2049>

云端 api 接口的使用简介:

打开 url: <https://api.diacloudsolutions.com.cn/>

由于该接口目前没有还未添加到 API list，所以随便选中一个，然后手动填写 URL

假设选择 api list 的第一个 **GET /devices**，选择右侧的“basic auth”



把 URL 更改为 <https://api.diacloudsolutions.com.cn/regs?start=2048&end=2049>  
输入在 diacloud 上注册的云账号，点击【Refresh headers】，将会出现 headers 的内容。

Instructions Basic Auth

diacloud@163.com

.....

Refresh headers

---

https://api.diacloudsolutions.com.cn/regs?start=2048&end=2049 GET send Headers URL param

---

request headers

Accept	application/json
Content-Type	application/json
Authorization	Basic ZGhhY2xvdWRAMTYzLmNvbTp0ZXN0MTIzNDU2

点击 send 后会出现如下信息：

```
Result:
{
  "data": {
    "21940": [
      {
        "device_id": "21940",
        "addr": 2049,
        "value": 157,
        "time": "2017-02-09 22:47:56",
        "name": "输入功率",
        "template": "return val/10+\"W\";",
        "history": 1,
        "device_name": "曲靖污水厂"
      }
    ],
    "23639": [
      {
        "device_id": "23639",
        "addr": 2048,
        "value": 0,
        "time": "2017-06-20 10:43:50",
        "name": "温度",
        "template": "return val+\"C\";",
        "history": 1,
        "device_name": "永丰污水厂"
      },
      {
        "device_id": "23639",
        "addr": 2049,
        "value": 0,
        "time": "2017-06-20 10:43:50",
        "name": "湿度",
        "template": "return val+\" %RH\";",
        "history": 1,
        "device_name": "永丰污水厂"
      }
    ]
  }
}
```

上图中 data 的中数据结构为

```
"device_id": [
    {
        "device_id": ,
```

```

"addr": 2048,

"value": 0,

"time": "2017-06-20 10:43:50",

"name": "温度",

"template": "return val+\`°C\`;",

"history": 1,

"device_name": "永丰污水厂"

}

```

data 中条目	名称	说明
Device_id	设备 id	设备在云平台的唯一标识
addr	地址	寄存器的编号
value	值	无符号十六进制
time	时间	寄存器上传的时间戳。该时间的格式为 UTC/GMT+08:00（东八区时间）。
name	寄存器名	寄存器的别名
template	转换脚本	在云端输入的转换脚本
History	是否保存历史	1、保存历史，0：不保存历史
Device_name	设备名	设备的别名
traffic_state	云流量状态	设备所用云流量状态 1:正常，0：用完

## 1.11 按 UTC 获取全部设备 alarm 信息接口

用于获取该账号下的设备的全部的 alarms 信息。

**注意：**该 Alarm 指的是在设备网页中设置的告警。详设置在设备网页中，【系统设置】->【事件管理】，事件类型中选择“告警事件”。



Json 请求格式:

url: <https://api.diacloudsolutions.com.cn/v2/alarms>

方法: get

请求头:

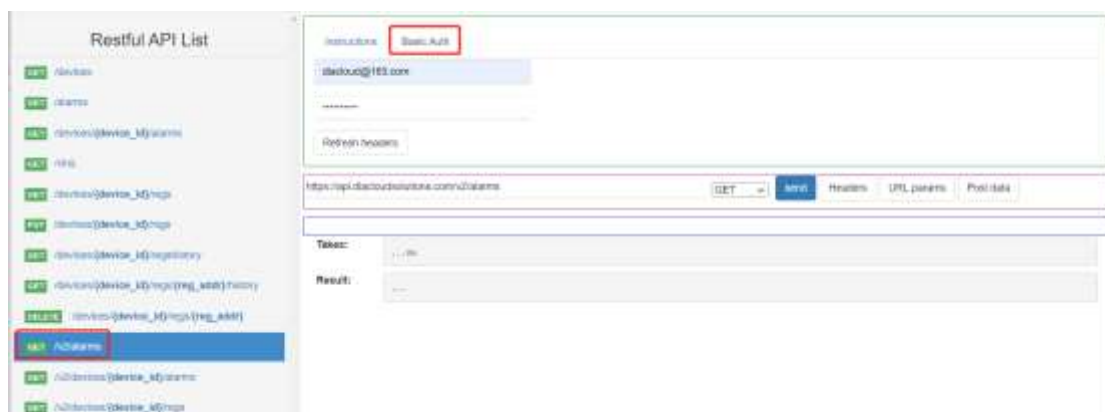
request headers	
Accept	application/json
Content-Type	application/json
Authorization	Basic ZGIhY2xvdWRAMTYzLmNvbTp0ZXN0MTIzNDU2

注: authoriz 为认证, 后面是 ( “用户名” : ” 密码” ) 的 base64 加密结果。

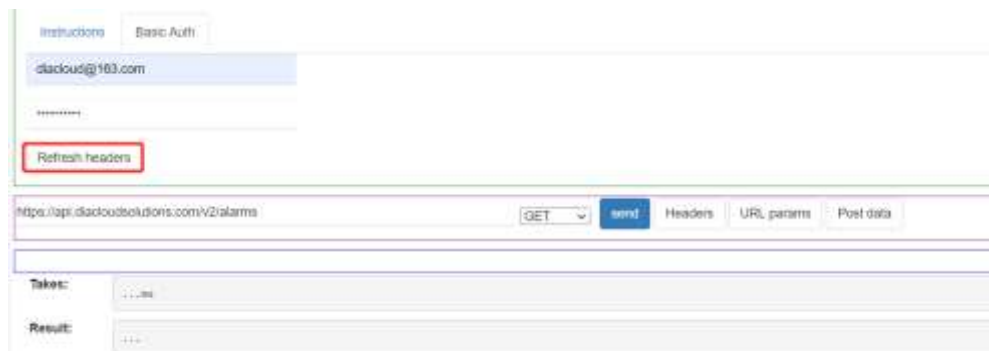
云端 api 接口的使用简介:

打开 url: <https://api.diacloudsolutions.com.cn/>

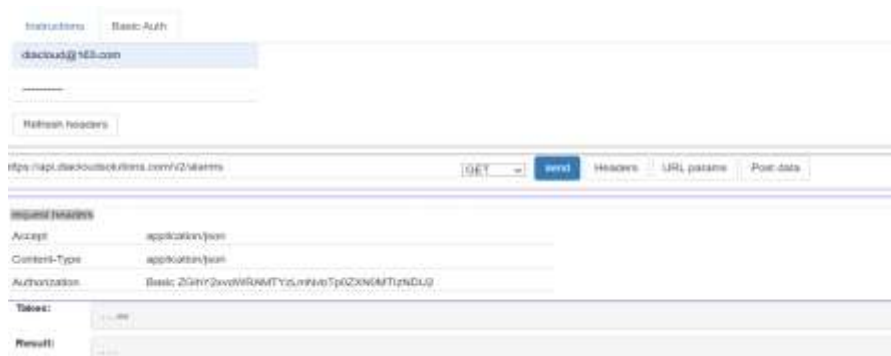
选择 api list 中的第二个 **GET /v2/alarms**, 选择右侧的 “basic auth”



输入在 diacloud 上注册的云账号, 点击【Refresh headers】



将会生成 request headers,



点击【Send】后，将会显示返回的 json:

```

Takes: 266 ms
Result:
{
  "paging": {
    "page": 1,
    "limit": 20,
    "total": 978,
    "pageCount": 49,
    "offset": 0
  },
  "data": [
    {
      "device_id": "22844",
      "name": "offline",
      "title": "offline",
      "content": "2017/01/04 14:38:18 offline offline",
      "sms": null,
      "email": {
        "result": 0,
        "addr": "TD{1160260354@qq.com}"
      },
      "created": "2017-01-04T06:38:28.127505Z"
    },
    {
      "device_id": "22844",
      "name": "offline",
      "title": "offline",
      "content": "2017/01/04 14:41:59 offline offline",
      "sms": null,
      "email": {
        "result": 0,
        "addr": "TD{1160260354@qq.com}"
      },
      "created": "2017-01-04T06:42:14.465561Z"
    }
  ]
}

```

其中返回请求包含两部分

### 1、Paging 分页信息

paging 中条目	名称	说明
page	第几页	
limit	每页多少条	
total	一共有多少条告警	
pagecount	一共有多少页	
offset	偏移	

### 2、Data 中的含义

data 中条目	名称	说明
device_id	设备 id	发生告警的设备的 ID
name	设备名	短信告警的名称
title	告警标题	告警标题
content	告警内容	告警内容
sms.result	短信发送状态	短信是否发送成功 0:成功 1 是失败（使用云短信时才有记录）
sms.addr	短信发送的手机号	短信发送的目的手机号
email.result	邮件发送结果	邮件是否发送成功 0:成功 1 是失败（使用云短信时才有记录）
email.addr	邮件发送地址	邮件发送的目的邮箱
created	发送时间	告警发送的时间。该时间的格式为 UTC 时间（0 时区时间）

注：告警支持分页、按时间查询等方式



如果查询第二页的话则

<https://api.diacloudsolutions.com.cn/v2/alarms?page=2>

如果查询 2018-01-01 到 2018-01-08

<https://api.diacloudsolutions.com.cn/v2/alarms?from=2018-01-01 00:00:00 & to=2018-01-09 00:00>

多个条件之间用&连接

## 1.12 按 UTC 获取单台设备 alarm 信息接口

用于获取该账号下的某台设备的全部的 alarms 信息。

注意：告警来源和 [1.11 按 UTC 获取全部设备 alarm 信息接口](#) 中的告警均来自设备网页中的设置的告警。具体设置参见 [1.10 按 UTC 获取全部设备 alarm 信息接口](#)。

Json 请求格式：

url: [https://api.diacloudsolutions.com.cn/v2/devices/{device\\_id}/alarms](https://api.diacloudsolutions.com.cn/v2/devices/{device_id}/alarms)

方法: get

请求头:

request headers	
Accept	application/json
Content-Type	application/json
Authorization	Basic ZGIhY2xvdWRAMTYzLmNvbTp0ZXN0MTIzNDU2

注: authoriz 为认证, 后面是 ( “用户名” : ”密码” ) 的 base64 加密结果。

注: 其中 {device\_id} 请更换为设备信息接口中 data 中的 id。比如 data 中的 ID 为 21669

Takes: 232 ms

Result:

```
{
  "data": [
    {
      "id": 21669,
      "name": "厦门",
      "sn": "DXR02010F0F0030",
      "tunnel_id": 0,
      "storage_state": 1,
      "traffic_state": 1,
      "online": 0,
      "ip": "0.0.0.0"
    },
    {
      "id": 21907,
      "name": "厦门市湖里区污水厂",
      "sn": "DXR02010E320027",
      "tunnel_id": 1446,
      "storage_state": 1,
      "traffic_state": 1,
      "online": 0,
      "ip": "192.168.199.180"
    },
    {
      "id": 21940,
      "name": "曲靖污水厂",
      "sn": "DXR02010E080006",
      "tunnel_id": 2373,

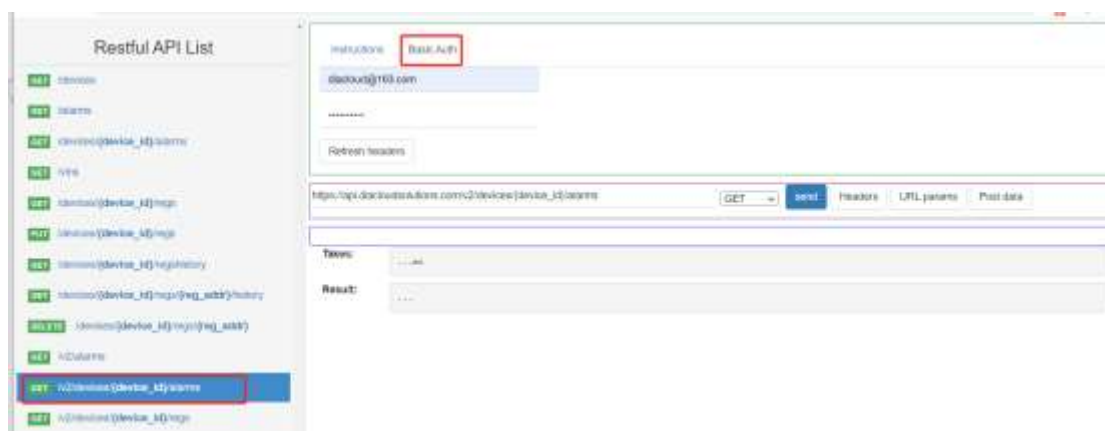
```

则 url 为 <https://api.diacloudsolutions.com.cn/v2/devices/21669/alarms>

云端 api 接口的使用简介:

打开 url: <https://api.diacloudsolutions.com.cn/>

选择 api list 中的第三个 **GET** `/v2/devices/{device_id}/alarms`, 选择右侧的“basic auth”, 并输入账号



在 url 中更换设备的 ID 号 21907 (有告警的设备 ID, 设备 ID 的查询方法请参看第一个基本信息接口)



点击 **【Send】** 后, 将会显示返回的 json:

103 ms

```
{
  "paging": {
    "page": 1,
    "limit": 20,
    "total": "38",
    "pageCount": 2,
    "offset": 0
  },
  "data": [
    {
      "device_id": "21907",
      "name": "online",
      "title": "online",
      "content": "online 2017/01/04 17:10:22",
      "sms": {
        "result": 0,
        "addr": "13616052915"
      },
      "email": {
        "result": 0,
        "addr": "TO:{1160260354@qq.com;}"
      },
      "created": "2017-01-04 17:10:56.410524"
    },
    {
      "device_id": "21907",
      "name": "online",
      "title": "online",

```

其中返回请求包含两部分

### 3、Paging 分页信息

paging 中条目	名称	说明
page	第几页	
limt	每页多少条	
total	一共有多少条告警	
pagecount	一共有多少页	
offset	偏移	

### 4、Data 中的含义

data 中条目	名称	说明
device_id	设备 id	发生告警的设备的 ID
name	设备名	短信告警的名称
title	告警标题	告警标题
content	告警内容	告警内容
sms.result	短信发送状态	短信是否发送成功 0:成功 1 是失败（使用云短信时才有记录）
sms.addr	短信发送的手机号	短信发送的目的手机号
email.result	邮件发送结果	邮件是否发送成功 0:成功 1 是失败（使用云短信时才有记录）
email.addr	邮件发送地址	邮件发送的目的邮箱
created	发送时间	告警发送的时间。该时间的格式为 UTC 时钟（0 时区时间）

注：告警支持分页、按时间查询等方式

如果查询第二页的话则

<https://api.diacloudsolutions.com.cn/v2/devices/21097/alarms?page=2>

如果查询 2018-01-01 到 2018-01-08

<https://api.diacloudsolutions.com.cn/v2/devices/21097/alarms?from=2018-01-01 00:00:00 &to=2018-01-09 00:00>

多个条件之间用&连接.

## 1.13 按 UTC 获取设备寄存器当前数据接口

用于获取该账号下的特定设备当前寄存器信息

Json 请求格式:

url: [https://api.diacloudsolutions.com.cn/v2/devices/{device\\_id}/regs](https://api.diacloudsolutions.com.cn/v2/devices/{device_id}/regs)

方法: get

请求头:

request headers	
Accept	application/json
Content-Type	application/json
Authorization	Basic ZGIhY2xvdWRAMTYzLmNvbTp0ZXN0MTizNDU2

注: authoriz 为认证, 后面是 ( “用户名” : ” 密码” ) 的 base64 加密结果。

注: 其中 {device\_id} 请更换为设备信息接口中 data 中的 id。比如 data 中的 ID 为 21669

Takes: 232 ms

Result:

```
{
  "data": [
    {
      "id": 21669,
      "name": "厦门",
      "sn": "DXR02010F0F0030",
      "tunnel_id": 0,
      "storage_state": 1,
      "traffic_state": 1,
      "online": 0,
      "ip": "0.0.0.0"
    },
    {
      "id": 21907,
      "name": "厦门市湖里区污水厂",
      "sn": "DXR02010E320027",
      "tunnel_id": 1446,
      "storage_state": 1,
      "traffic_state": 1,
      "online": 0,
      "ip": "192.168.199.100"
    },
    {
      "id": 21940,
      "name": "曲靖污水厂",
      "sn": "DXR02010E000006",
      "tunnel_id": 2373,

```

则 url 为 <https://api.diacloudsolutions.com.cn/devices/21669/regs>

注 1: 默认返回数据从 2048 开始, 如果需要读取 bit 区域数据 ( \$1536~\$2047 ) 需要添加 start 参数, 如:

<https://api.diacloudsolutions.com.cn/devices/21669/regs?start=1536>.

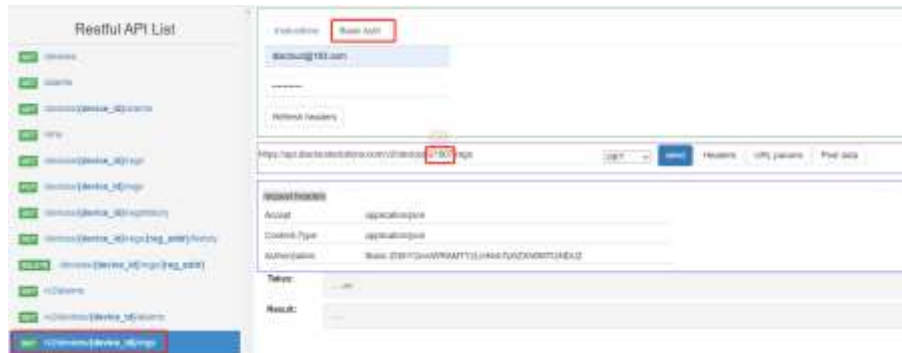
注 2: 如果是账号错误或者设备 ID 错误将返回 403 error。

注 3: 该接口不可以用于获取 \$1536 之前系统寄存器的数据。

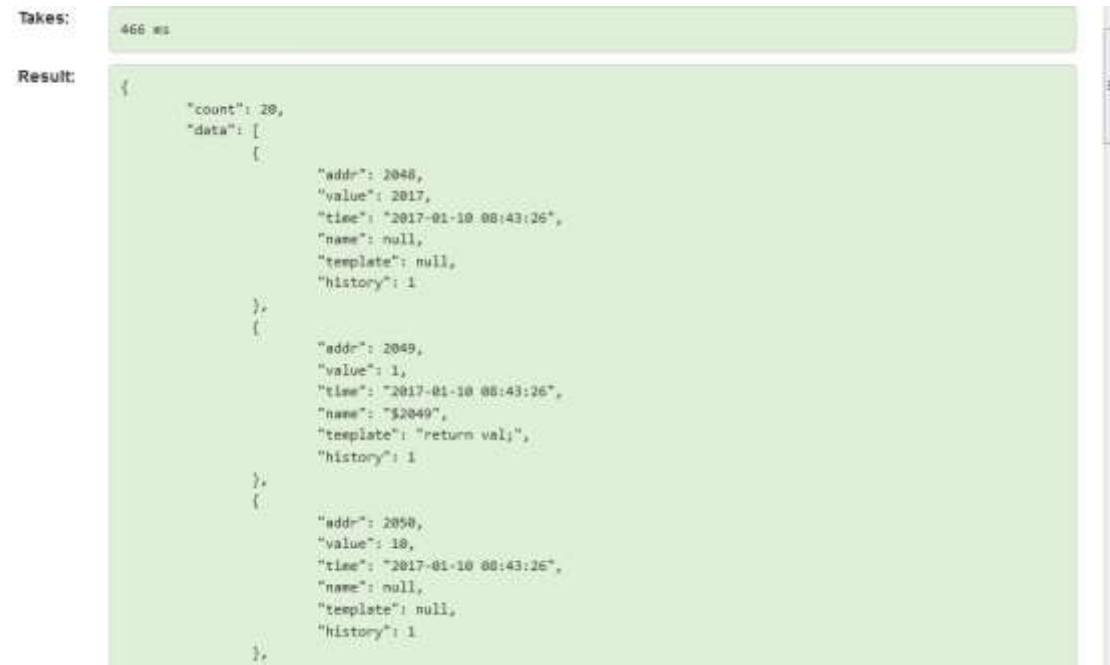
## 云端 api 接口的使用简介:

打开 url: <https://api.diacloudsolutions.com.cn/>

选择 api list 中, **GET /v2/devices/{device\_id}/regs**, 选择右侧的“basic auth”



填写完账号和密码后, 填入设备 ID 后, 点击【Send】后, 将会显示返回的 json:

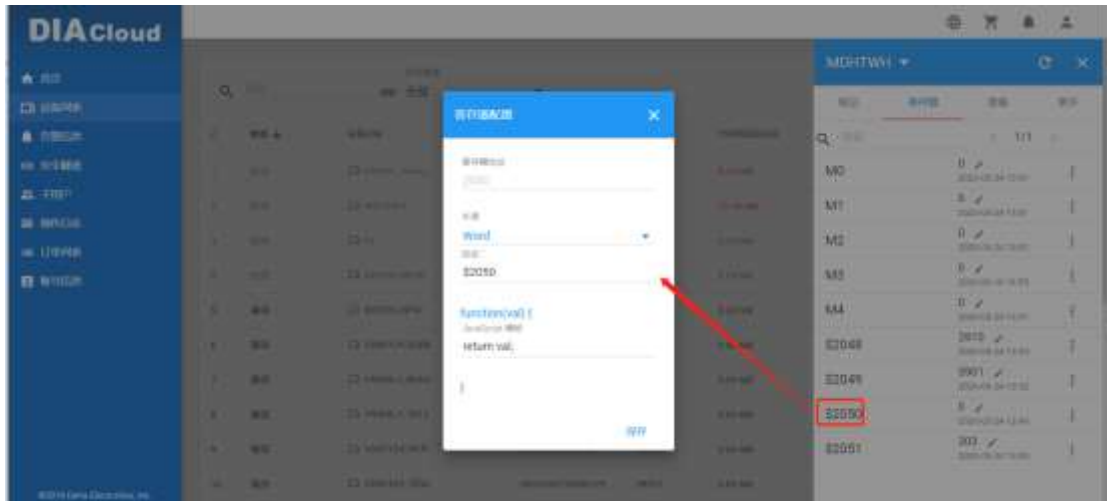


响应分成两个部分, count 和 data, 中 count 是寄存器的个数。

其中 data 涉及到字段说明如下:

data 中条目	名称	说明
Addr	寄存器地址	寄存器地址 (2048~4096)
Value	寄存器值	寄存器值 (无符号十进制整数, 其他类型需要自己转换)
Time	时间	寄存器数值上传的时间。该时间的格式为 UTC 时钟 (0 时区)。
templatet	转换脚本	原始值需要加工转后显示的脚本
History	是否保存历史	1、保存历史, 0: 不保存历史

其中 name 的属性需要在 <https://api.diacloudsolutions.com.cn/> 的网站上【设备列表】中, 选中该台设备, 最右边的操作【更多】->【寄存器】, 最右边更多的【配置】功能, 对应的别名。

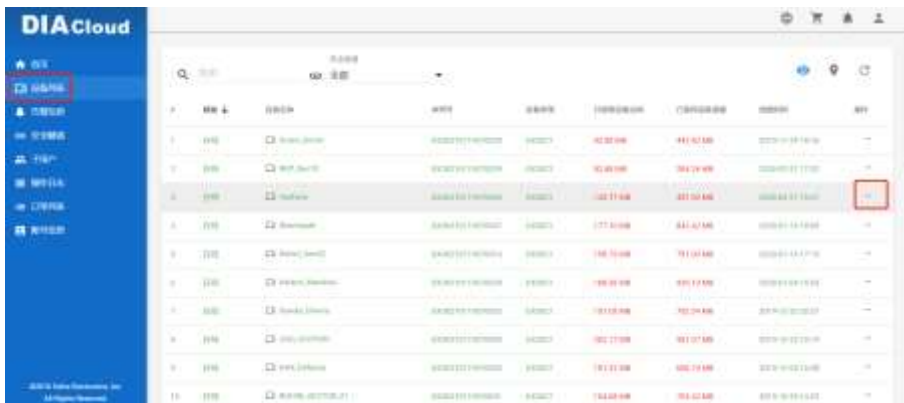


## 1.14 按寄存器配置获取历史数据接口

用户可以在网站上配置寄存器的类型，通过该接口后台自动将数据组合好后返回。

### 1.14.1 寄存器的配置步骤

登陆 <https://diacloudsolutions.com.cn/> 网站，【设备列表】中操作 ...



选中【寄存器】页面中的 ... -> 【配置】



可以对长度进行更改，比如 float、dword、word。



### 1.14.2 按寄存器配置获取历史数据接口介绍

Json 请求格式:

url: [https://api.diacloudsolutions.com.cn/devices/{device\\_id}/regshistory?regs={reg\\_addr1},{reg\\_addr2}](https://api.diacloudsolutions.com.cn/devices/{device_id}/regshistory?regs={reg_addr1},{reg_addr2})

请求参数:

regs: **必填**。格式样例: ?regs=:2048,2050

page: 可选

limit: 每页返回数量, 可选

from: 查询历史的开始时间参数, 可选。时间格式为 UTC 时钟, 比如: 2019-01-23 05:40:48。

to: 查询历史的结束时间参数, 可选。时间格式为 UTC 时钟, 比如: 2019-01-23 05:40:48。

gap: 固定的时间间隔, 单位为秒。如果该参数缺省, 默认返回的时间间隔为 60 秒; 如果该参数填了后, 数据就会按设定的间隔返回数据。

方法: get

请求头:

request headers	
Accept	application/json
Content-Type	application/json
Authorization	Basic ZGIhY2xvdWRAMTYzLmNvbTp0ZXN0MTIzNDU2

注: authoriz 为认证, 后面是 ( “用户名” : ” 密码” ) 的 base64 加密结果。

注: 其中 {device\_id} 请更换为设备信息接口中 data 中的 id。比如 data 中的 ID 为 21669

注: 其中 {reg\_addr} 寄存器的地址

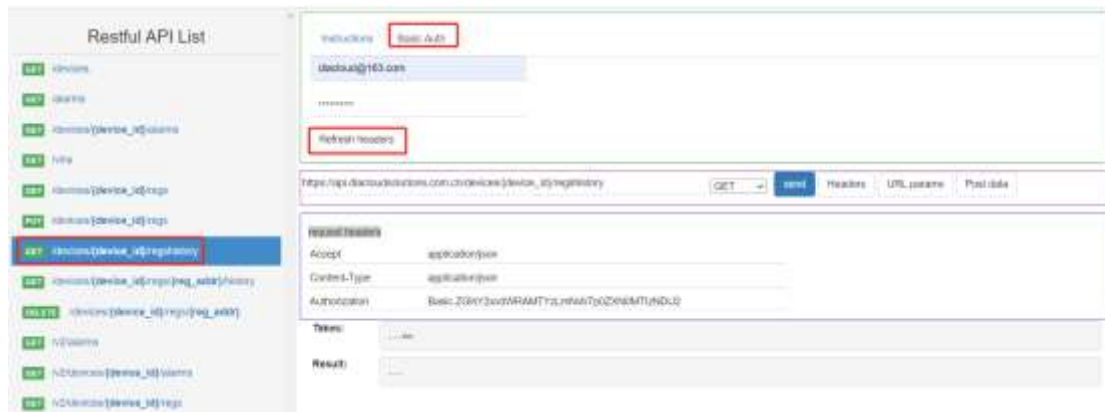
假设需要获取设备 ID 为 35236 的 2048 和 2050 寄存器的历史数据则 url 为

<https://api.diacloudsolutions.com.cn/devices/35236/regshistory?regs=2048,2050>

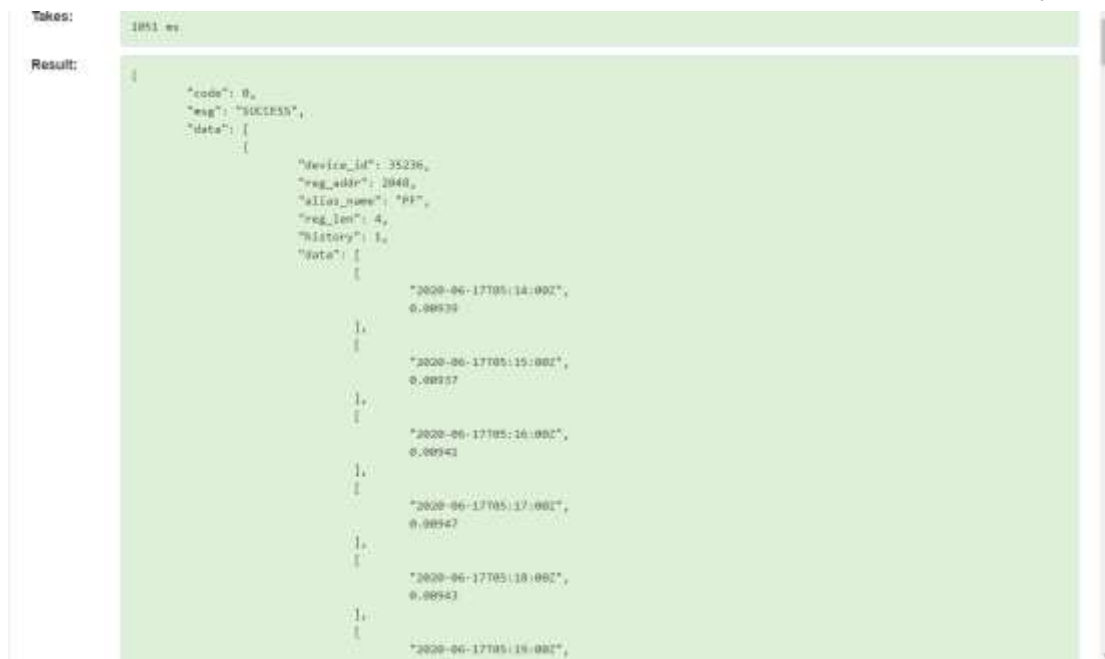
云端 api 接口的使用简介:

打开 url: <https://api.diacloudsolutions.com.cn/>

选择 api list 中的,选择右侧的“basic auth”



填写完账号和密码后,填入设备 ID 后和寄存器编号,点击【Send】后,将会显示返回的 json:



响应分成三个部分,其中 code 和 msg, 表示请求状态。

code:0 表示成功,非 0 表示失败。

msg:success, fail。

当调用数据没有权限的时候将会返回失败。

外层 data 中,由 6 个部分构成

外 data 中条目	名称	说明
device_id	设备 ID	
reg_addr	寄存器地址	寄存器编号
alias_name	寄存器别名	寄存器配置中配置的别名
reg_len	寄存器长度	请求的寄存器个数。(由于 2048 是 float, 2050 是 Dword, 所以一共 4 个寄存器)
history	是否保存历史	1、保存历史, 0: 不保存历史
data	内 data	返回的历史数据就在其中

内 data 涉及到字段说明如下:



```

"device_id": 35236,
"reg_addr": 2048,
"alias_name": "PF",
"reg_len": 4,
"history": 1,
"data": [
  [
    "2020-06-17T05:40:00Z",
    0.00949
  ],
  [
    "2020-06-17T05:41:00Z",
    0.00936
  ],
  [
    "2020-06-17T05:42:00Z",
    0.00947
  ],
  [
    "2020-06-17T05:43:00Z",
    0.00942
  ],
  [
    "2020-06-17T05:44:00Z",
    0.00948
  ],
  [
    "2020-06-17T05:45:00Z",
    0.00941
  ],
  [
    "2020-06-17T05:46:00Z",

```

内 data 中条目	值	说明
reg_time	2020-06-17T05:45:00Z	寄存器上传的时间，含义为 2020 年，6 月 17 日 05:24:26, 时区为 0 时区
Device_id	35236	设备 ID
Reg_addr	2048	寄存器的编号为 2048
寄存器值	0.00949	寄存器的值为 0.00949

注：寄存器支持分页、按时间查询等方式

如果查询 2020-06-16 到 2020-06-17

<https://api.diacloudsolutions.com.cn/devices/35236/regshistory?regs=2048,2050&from=2020-06-16 00:00:00 & to=2020-06-17 00:00:00>

如果 from 和 to 的参数均缺省的情况，默认返回最近一小时内的数据，数据最多 2000 条

如果按等时间间隔，的请求参数如下。查询 2020 年 2020-06-16 一天的数据，固定间隔为 90 秒。

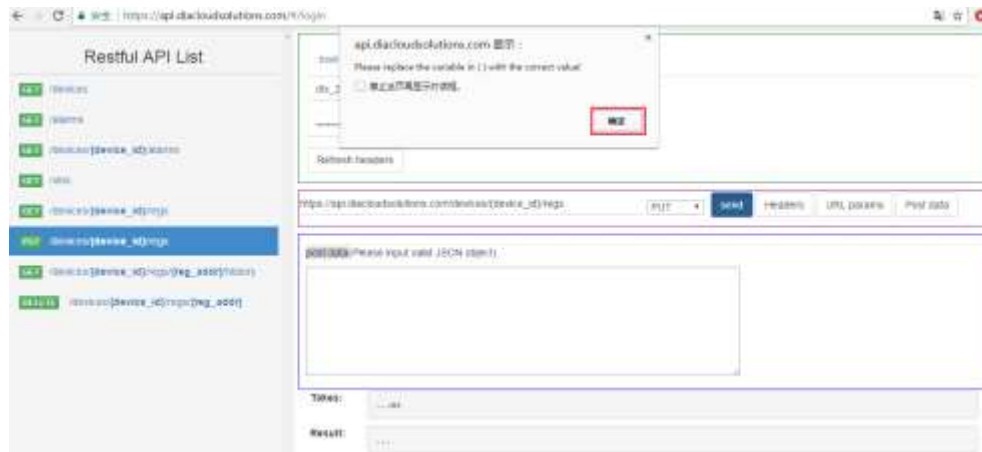
<https://api.diacloudsolutions.com.cn/devices/35236/regshistory?regs=2048,2050&from=2020-06-16 00:00:00 & to=2020-06-17 00:00:00&gap=90>

ps: 单次调用历史数据，返回条数最多 1 万条。

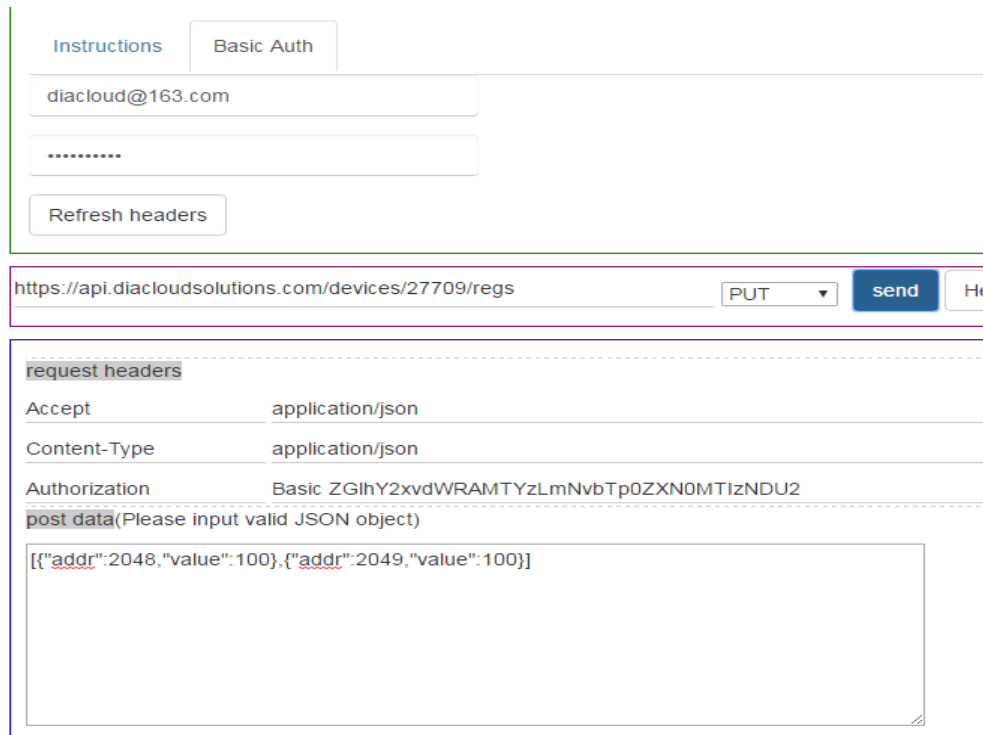
# FAQ: 常见问题

## F1: 调用 API 提示“please replace the variable in {} with the correct value”

调用修改寄存器 Api 的时候提示: “please replace the variable in {} with the correct value”?  
提示如下图:



这是因为修改值的时候需要在 post data 中添加需要修改的内容如下图所示, 并且修改内容必须是已经上传云的寄存器。



## F2: 调用 API 返回 403fobind

当返回错误提示 403 的时候，请检查用查询信息和 URL 是否正确。

## F3: 显示设置成功，但没有生效

设置寄存器数据的时候，其返回的成功，是表示服务器收到数据，并向设备端转发。但是设备端有没有执行成功，有没有收到信息无法确认的。客户需要自己通过获取寄存器值的接口进行查询确认（一般情况下，5 内数据没有变化为需要显示的值之后，需要进行重试）。设置不成功一般会有以下几个问题：

- (1) 设备网络不稳定，经常发生网络重拨，或者信号不好
- (2) 设备端数据为只读属性
- (3) 设备端数据取消了上传云。

## F4: Dword 和浮点型的算法

API 获取的是原始的寄存器的原始值，其为 16 位无符号整形。在常规的使用会经常用到 32 整形和浮点型两种显示显示。其由两个相邻寄存器的值组合而成。假设我们要组合值，寄存器 1（\$2048）的值 51590,寄存器 2（\$2049）的值 16268。

其中 Dword: dword 的值=寄存器 2 的值\*65536+寄存器 1 的值=16268\*65536+51590

浮点型: 计算需要通过 dword 的结果进行转换得到浮点型的数据

Js 脚本范例: convert 为将 float 转换浮点数的函数

```
> function convert(val) {
  var sign = (val & 0x80000000)?-1:1;
  var mantissa = (val & 0x7FFFFFFF)|0x800000;
  var exp = ((val>>23) & 0xFF)-127-23;
  var floatval = sign*mantissa*Math.pow(2,exp);
  return floatval.toFixed(3);
}

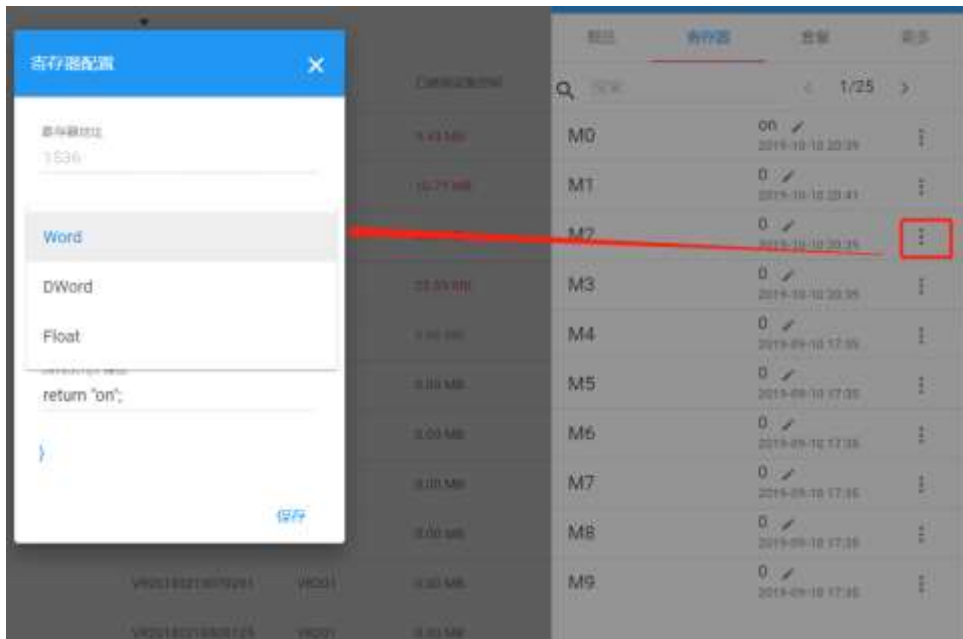
var a = 16268*65536+ 51590;
convert(a)
< "1.100"
>
```

Ps:a 为 dword 的结果。

用户在调试的时候，为了方便核对，可以登录 [www.diacloudsolutions.com.cn](http://www.diacloudsolutions.com.cn) 网站，【设备列表】中“操作”->“更多”，在弹窗中，选择寄存器的 table。



在寄存器中的更多可以配置，寄存器的显示类型，来核对自己写的转换函数。



PS: 这里的配置的类型，只是前端解析脚本，并不会影响 API 返回的值。

## F5: 设备寄存器当前数据接口，获取不到 M 区数据?

设备 M 区域规划为寄存器\$1536 到\$2047，也就是 M0 其实寄存器编号是\$1536，M1 是\$1537 以此类推到 M511 是 2047。若要获取到 M 区域，在需要加检索参数?start=1536，示例如下：  
<https://api.diacloudsolutions.com.cn/devices/21669/regs?start=1536>.

## F6: 设备寄存器当前数据接口，获取数据不全?

获取设备寄存器当前数据接口，默认开始参数是从 2048 开始。如果有用到 bit 区域，显示如下，有显示 M0 等，其以 M 开头的就是 M 区域。所以不是数据不全，而是 start 参数不对。如何获取，M 区数据，参看 [F5 获取不到寄存器 M 区域的数据](#)。



## F7: API 调用限制

API 的调用, 无论是否是同一个源 IP, 只要是同一个账号, 每小时调用次数不能超过 10000 次 (第一章节中的每个 API 接口调用一次, 算一次调用), 否则可能会被限制该账号在一个小时内无法调用。

同时 API 的调用并不是免费的, 其对应具体费用请咨询对应供货渠道。

## F8: 如何上传通讯错误标志

通讯错误相关寄存器信息如下:

寄存器	modbus 地址		说明	支持机型
	DEC	HEX		
\$900	900	384	RS232 Modbus 主站通讯错误条数	DX2100/DX2300 (FW1.5 之后版本)
\$901	901	385	RS232 Modbus 第一条错误所在位置	DX2100/DX2300 (FW1.5 之后版本)
\$902	902	386	RS232 Modbus 第一条错误代码	DX2100/DX2300 (FW1.5 之后版本)
\$903	903	387	RS485 Modbus 主站错误条数	DX2100/DX2300 (FW1.5 之后版本)
\$904	904	388	RS485 Modbus 第一条错误所在位置	DX2100/DX2300 (FW1.5 之后版本)
\$905	905	389	RS485 Modbus 第一条错误代码	DX2100/DX2300 (FW1.5 之后版本)
\$906	906	38A	Modbus TCP 错误条数	DX2100/DX2300 (FW1.5 之后版本)
\$907	907	38B	Modbus TCP 第一条错误所在组	DX2100/DX2300 (FW1.5 之后版本)
\$908	908	38C	ModbusTCP 第一条错误所在组中相对位置	DX2100/DX2300 (FW1.5 之后版本)
\$909	909	38D	Modbus TCP 第一条错误代码	DX2100/DX2300 (FW1.5 之后版本)
\$910	910	38E	综合指示 Modbus TCP 和 SiemensTCP 通讯状态 (0 为正常, 1 为错误)	DX2100/DX2300 (FW1.5 之后版本)
\$911	911	38F	Siemens TCP 错误条数	DX2100/DX2300 (FW1.5 之后版本)
\$912	912	390	Siemens TCP 第一条错误所在组	DX2100/DX2300 (FW1.5 之后版本)

\$913	913	391	Siemens TCP 第一条错误所在组中相对位置	DX2100/DX2300 (FW1.5 之后版本)
-------	-----	-----	---------------------------	-------------------------------

由于 api 无法获取到通讯状态，所以需要通过 modbus tcp 读取上面的寄存器数据，然后上传到云端。比如需要上传 485 的通讯状态。485 的通讯状态相关寄存器有主站错误调试 903，第一条所在位置 904，第一条对应的错误码 905。

设备网页中，【系统设置】->【Modbus TCP】，选择【Modbus TCP 服务端+客户端】，点击【添加服务器】



分别按下图填写对应的信息。下图是将 903 之后的寄存器关联到 2200.设置完成后，点击【保存】



【系统设置】->【寄存器管理】，点击【添加】



按下图设置完成后，点击【保存】，之后就可以通过 api 调取 2200 寄存器的值，而得到 PLC 的通讯状态。

寄存器管理 添加/删除路由器的寄存器。

系统设置 > 寄存器管理

添加

寄存器类型	Word
寄存器地址	\$2200 (\$2048-4095, M0-511)
长度	3
是否上云	是
是否保存历史记录	否

保存

返回