

文件编号:	
密级:	
项目 ID:	
总页数:	

Gate6 云平台接口

产品接入规范

版本<1.0.1>

拟制	_____	日期	2016-04-12
审核	_____	日期	_____
批准	_____	日期	_____



珠海东帆科技有限公司

声明

修改记录

版本	日期	AMD	修订者	说明
1.0.0	20160225		陈智勇	建立
1.0.1	20160301		邹晓明	完善程序框架图 and 提供接口
1.0.2	20160412		邹晓明	精简部分内容作为客户版本
1.0.3	20160422		邹晓明	增加下发文件的接口。网关通过实时数据的返回来得知有下发文件的。然后网关通过下发文件接口，获取下发的文件信息，下发具体的文件，通知服务器接收完成。
1.04	20160712		邹晓明	增加整体流程图
1.05	20161222		邹晓明	增加压缩选项的说明
1.06	20191217		王锋 温天榕	增加上位机开发说明 增加 name 解析

1 文档介绍

1.1 文档目的

编写此文档的目的是为了明确云平台与产品之间的接口调用规范。

1.2 文档范围

本文档系统阐述了所有使用到的接口

1.3 读者对象

- 云平台开发人员
- 产品开发人员

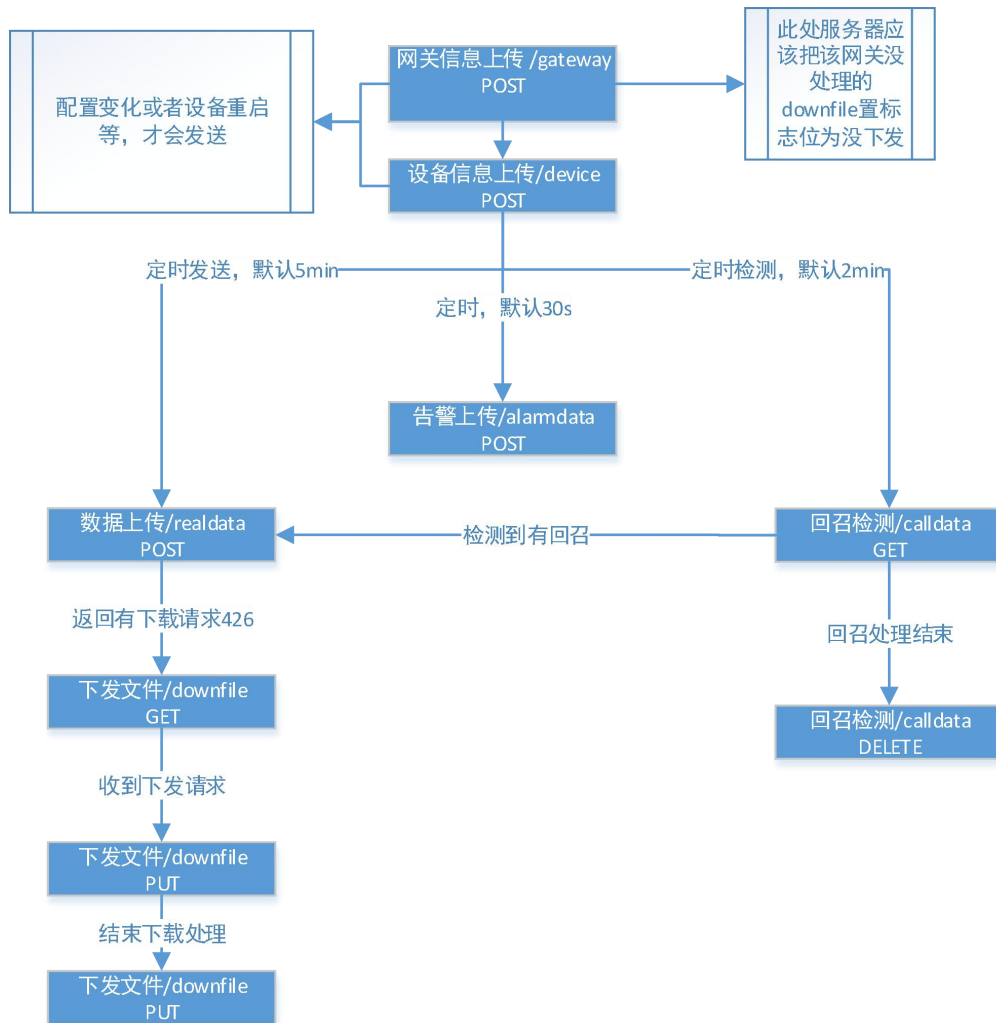
1.4 术语与缩写解释

缩写、术语	解释

2 系统概述

云平台涉及存储与分析一体的大数据处理平台，通过接收设备上传的数据，对数据进行建模分析。

整体流程图如下：



3 接口要求

3.1 框架要求

为适应产品的通用性，本文档适配 JSON 和 XML 格式传输，本框架使用 RESTFUL 框架。

3.2 安全要求

3.3 共同约定

编码格式采用 UTF-8 格式。

3.4 数据类型

采用 JSON 传递参数和返回结果，数据类型：

String：字符、数字，不包含汉字

NString：汉字、字符、数字，utf-8 编码

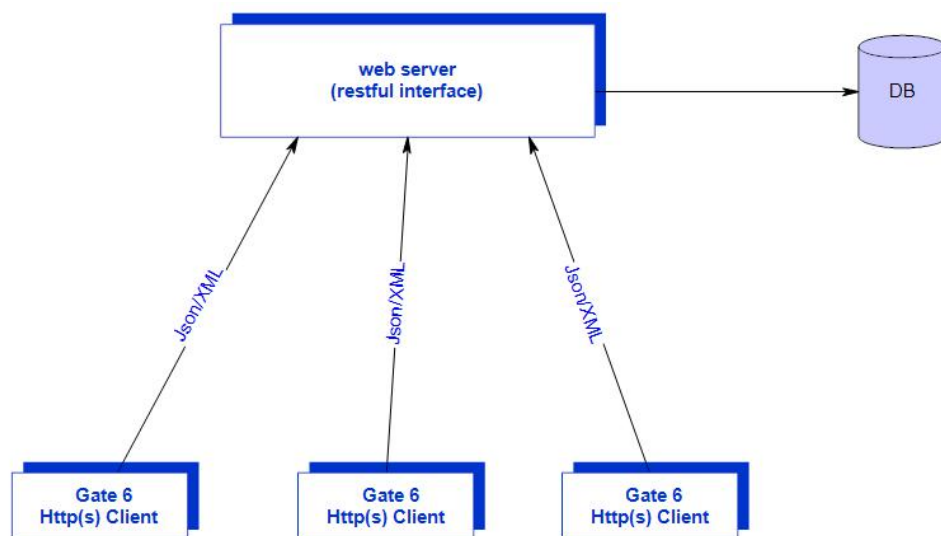
Number：只能是数字

4 程序框架

4.1 说明

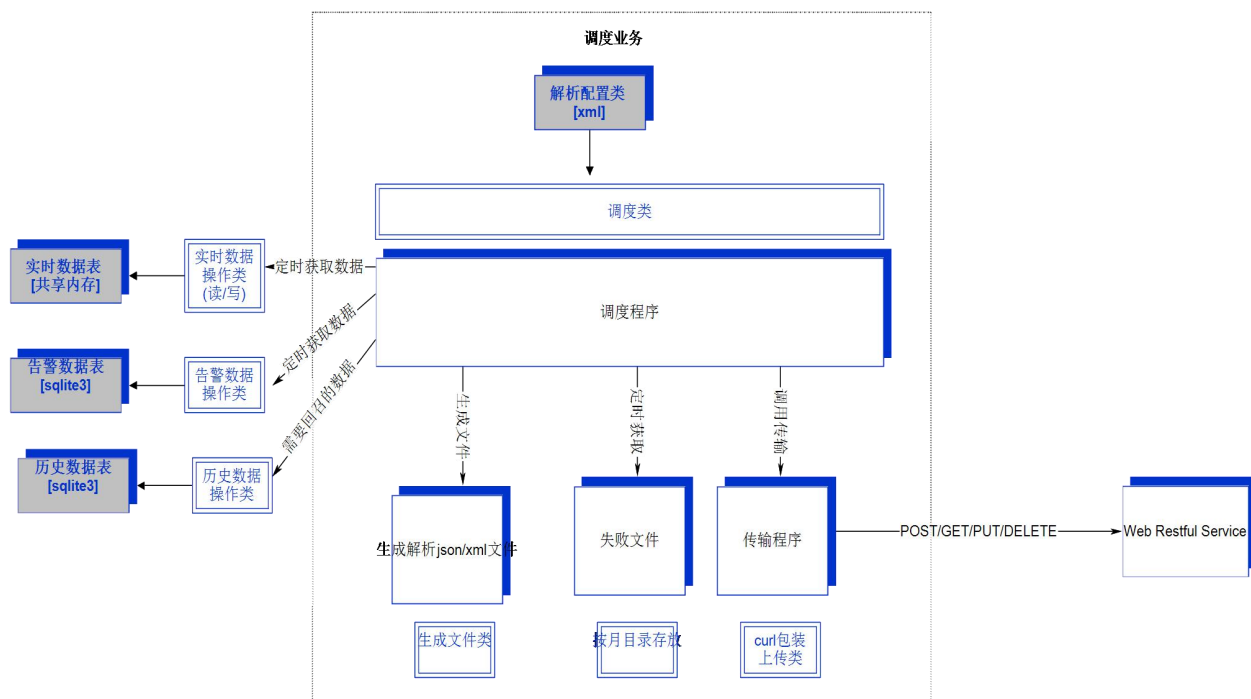
1. 通用上传，restful 方式
基本是给客户简单的上传，并可以提供服务端的 php demo 给客户。
2. 整理统一方式，避免每个定制零散
3. 本上传方式提供了压缩选项，压缩算法采用 deflate。主要应用场景为采集点多，需要上传的内容较多的情况。如果是采用压缩方式，http 上传的头部，会有选项“Content-Encoding: deflate”。服务端收到数据后，先要进行解压，然后和原来的处理一样。目前提供的 xampp.7z 绿色演示 demo，已经兼容压缩和非压缩的上传。

4.2 系统框架图



Gate6 作为 Http Client,向 webservice 提交 Json/XML 流。

4.3 模块框架图



如上图所示：

实时数据操作类：负责对共享内存的实时数据进行读写；

告警数据操作类：负责对 sqlite3 中的告警数据进行读写；

历史数据类：负责读写历史数据，查询指定时间的记录。

配置解析类：解析网关、仪表信息，解析实时数据点配置，解析服务器配置，服务器配置的选项有如下：

字段	说明
url	上传的 url 基本前缀，如 http://192.168.13.64:8080/test ; 则网关创建接口为 http://192.168.13.64:8080/test/gateway
https	是否是 https 上传
certfile	证书（https 有效）
realInterval	上传实时数据时间间隔
alarmInterval	上传告警数据的间隔

uploadtype	xml/json
------------	----------

生成文件类：提供生成文件（网关接口文件，仪表接口文件，实时数据文件，实时告警文件）接口，供调度类调用生成对应的文件；

//回召

调度类：负责按照设定好的上传时间规则，定时调用实时数据/告警数据类，获取所需的数据；然后调用生成文件类，生成上传的文件；调用 curl 包装上传类，进行 http/ftp/https 等的调用。

4.4 基本目录

4.5 网关接口

4.5.1 网关创建接口

4.5.1.1 接口定义

接口编号	0001	接口名称	/gateway
接口描述	网关创建接口		
调用方式	POST		
调用说明	网关创建。		
Header 参数	Content-Type:application/json	Content-Type:application/xml	
Body 参数	<pre>{ "sn": "GW1603019001", "name": "GateWay1", "ver": "1.0.0", "site": { "name": "Pilot", "longitude": "120.23", "latitude": "120.23" }, "owner": { "name": "User1", "desc": "description" }, "room": { "name": "Room1", "desc": "description",</pre>	<pre><?xml version="1.0" encoding="UTF-8"?> <data> <gw sn="GW1603019001" name="GateWay1" ver="1.0.0"> <site name="Pilot" longitude="120.23" latitude="120.23" /> <owner name="User1" desc="description" /> <room name="Room1" desc="description" camera="cameraserail"/> </gw> </data></pre>	

	<pre>"camera": "camera" } }</pre>			
调用举例	http://192.168.13.64:80/gateway			
参数说明				
参数名称	类型	长度	必填	说明
sn	String	36	必填	序列号
name	String	128	必填	网关名字
ver	String	10	必填	网关软件版本号
site	节点			站点信息
name	String	128	必填	站点名称
longitude	String	10		经度
latitude	String	10		纬度
owner	节点			业主信息
name	String	128	必填	业主名称
desc	String	128		业主描述
room	节点			机房信息
name	String	128	必填	机房名称
desc	String	128		机房描述
camera	String	128		摄像头编号
返回数据				
参数名称	类型	长度	必填	说明

4.5.1.2 调用举例:

接口	/gateway	
(1)	提交申请, 返回值 http code 为 201 , 表示创建成功。http code 406, 表示创建信息失败。	
参数	<pre> { "sn": "GW1603019001", </pre>	<pre> <?xml version="1.0" encoding="UTF-8"?> <data> </pre>

值	<pre> "name": "GateWay1", "ver": "1.0.0", "site": { "name": "Pilot", "longitude": "120.23", "latitude": "120.23" }, "owner": { "name": "User1", "desc": "description" }, "room": { "name": "Room1", "desc": "description", "camera": "camera" } } </pre>	<pre> <gw sn="GW1603019001" name="GateWay1" ver="1.0.0"> <site name="Pilot" longitude="120.23" latitude="120.23" /> <owner name="User1" desc="description" /> <room name="Room1" desc="description" camera="1245689" /> </gw> </data> </pre>
返回	状态码返回： 201 Created 创建成功 406 创建的信息有错误	

4.5.2 查看网关信息接口

4.5.2.1 接口定义

接口编号	0002	接口名称	/gateway/<gateway_sn>		
接口描述	查看网关信息接口				
调用方式	GET				
调用说明	查看网关信息。				
Header 参数	Content-Type:application/json		Content-Type:application/xml		
Body 参数					
调用举例	http://192.168.13.64:80/gateway/GW1603019001				
参数说明					
参数名称	类型	长度	必填	说明	
返回数据					
参数名称	类型	长度	必填	说明	

sn	String	36	必填	序列号
name	String	128	必填	网关名字
ver	String	10	必填	网关软件版本号
site	节点			站点信息
name	String	128	必填	站点名称
longitude	String	10		经度
latitude	String	10		纬度
owner	节点			业主信息
name	String	128	必填	业主名称
desc	String	128		业主描述
room	节点			机房信息
name	String	128	必填	机房名称
desc	String	128		机房描述
camera_num	String	128		摄像头编号

4.5.1.2 调用举例:

接口	/gateway/GW1603019001	
(1)	提交申请, 返回值 http code 为 200 , 表示查询成功。http code 404, 表示查询网关不存在。	
参 数 值		
返 回	<pre>{ "sn": "GW1603019001", "name": "GateWay1", "ver": "1.0.0", "site": { "name": "Pilot", "longitude": "120.23", "latitude": "120.23" }, "owner": {</pre>	<pre><?xml version="1.0" encoding="UTF-8"?> <data> <gw sn="GW1603019001" name="GateWay1" ver="1.0.0"> <site name="Pilot" longitude="120.23" latitude="120.23" /> </gw> <owner name="User1" desc="description" /> <room name="Room1" desc="description" camera="1245689" /> </data></pre>

	<pre> "name": "User1", "desc": "description" }, "room": { "name": "Room1", "desc": "description", "camera_num": "12345689" }, } </pre>	
--	--	--

4.6 仪表接口

4.6.1 仪表创建接口

4.6.1.1 接口定义

接口编号	0003	接口名称	/device/{gateway_sn}		
接口描述	仪表创建接口				
调用方式	POST				
调用说明	仪表创建。				
Header 参数	Content-Type:application/json		Content-Type:application/xml		
Body 参数	<pre>{ "device": [{ "channel": "1", "id": "1", "name": "PMAC201HW", "desc": "desc dev1" }, { "channel": "1", "id": "2", "name": "PMAC201HW", "desc": "desc dev2" }, { "channel": "1", "id": "3", "name": "PMAC201HW", "desc": "desc dev1" }, ] }</pre>		<pre><?xml version="1.0" encoding="UTF-8"?> <data> <device channel="1" id="1" name="PMAC201HW" desc =" desc dev1" /> <device channel="1" id="2" name="PMAC201HW" desc =" desc dev1" /> </data></pre>		
调用举例	http://192.168.13.64:80/device/GW1603019001				
参数说明					
参数名称	类型	长度	必填	说明	
device	节点			注意：channel+id 可以作为同一网关下设备的唯一标志	

channel	String	10	必填	通道号
id	String	10	必填	设备通讯地址号
name	String	128	必填	设备名称
desc	String	128		设备描述
返回数据				
参数名称	类型	长度	必填	说明

4.6.1.2 调用举例:

接口	/device/GW1603019001		
(1)	提交申请， 返回值 http code 为 201 ，表示创建成功。http code 406,表示创建信息失败。		
参 数 值	<pre>{ "device":[{"channel":"1", "id":"1", "name":"PMAC201HW", "desc":"desc dev1"}, {"channel":"1", "id":"2", "name":"PMAC201HW", "desc":"desc dev2"}, {"channel":"1", "id":"3", "name":"PMAC201HW", "desc":"desc dev1"},] }</pre>	<pre><?xml version="1.0" encoding="UTF-8"?> <data> <device channel="1" id="1" name="PMAC201HW" desc =" desc dev1" /> <device channel="1" id="2" name="PMAC201HW" desc =" desc dev1" /> </data></pre>	
	返 回	状态码返回： 201 Created 创建成功 406 创建的信息有错误	

4.6.2 查看仪表信息接口

4.6.2.1 接口定义

接口编号	0004	接口名称	/device/<gateway_sn>
接口描述	查看网关下仪表信息接口		
调用方式	GET		
调用说明	查看网关下仪表信息。		
Header 参数	Content-Type:application/json	Content-Type:application/xml	
Body 参数			
调用举例	http://192.168.13.64:80/device/GW1603019001		
参数说明			

参数名称	类型	长度	必填	说明
返回数据				
参数名称	类型	长度	必填	说明
device	节点			注意：channel+id 可以作为同一网关下设备的唯一标志
channel	String	10	必填	通道号
id	String	10	必填	设备通讯地址号
name	String	128	必填	设备名称
desc	String	128		设备描述

4.6.2.2 调用举例：

接口	/gateway/GW1603019001	
(1)	提交申请， 返回值 http code 为 200 ， 表示查询成功。http code 404, 表示查询网关不存在。	
参 数 值		
返回	<pre>{ "device":[{"channel":"1", "id":"1", "name":"PMAC201HW", "desc":"desc dev1"}, {"channel":"1", "id":"2", "name":"PMAC201HW", "desc":"desc dev2"}, {"channel":"1", "id":"3", "name":"PMAC201HW", "desc":"desc dev1"},] }</pre>	<pre><?xml version="1.0" encoding="UTF-8"?> <data> <device channel="1" id="1" name="PMAC201HW" desc =" desc dev1" /> <device channel="1" id="2" name="PMAC201HW" desc =" desc dev1" /> </data></pre>

4.7 上传实时数据接口

4.7.1 网关上传实时接口

4.7.1.1 接口定义

接口编号	0005	接口名称	/realdata/<gateway_sn>
接口描述	网关实时数据上传接口		
调用方式	POST		
调用说明	网关实时数据上传。		
Header 参数	Content-Type:application/json	Content-Type:application/xml	

Body 参数

```
{
  "logdt": "2016-02-19 00: 05: 00",
  "device": [
    {
      "id": "1",
      "channel": "1",
      "node": [
        {
          "name": "Comm_Status",
          "value": "1",
          "unit": ""
        },
        {
          "name": "voltA",
          "value": "220.1",
          "unit": "V"
        },
        {
          "name": "voltB",
          "value": "219.8",
          "unit": "V"
        },
        {
          "name": "voltC",
          "value": "219.9",
          "unit": "V"
        },
        {
          "name": "currA",
          "value": "4.89",
          "unit": "A"
        },
        {
          "name": "currB",
          "value": "5.01",
          "unit": "A"
        },
        {
          "name": "currC",
          "value": "4.98",
          "unit": "A"
        }
      ]
    },
    {
      "id": "2",
      "channel": "1",
      "node": [
        {
          "name": "Comm_Status",
          "value": "1",
          "unit": ""
        },
        {
          "name": "voltA",
          "value": "220.1",
          "unit": "V"
        },
        {
          "name": "voltB",
          "value": "220.1",
          "unit": "V"
        },
        {
          "name": "voltC",
          "value": "220.1",
          "unit": "V"
        },
        {
          "name": "currA",
          "value": "5.01",
          "unit": "A"
        },
        {
          "name": "currB",
          "value": "5.01",
          "unit": "A"
        },
        {
          "name": "currC",
          "value": "5.01",
          "unit": "A"
        }
      ]
    }
  ]
}
```

<?xml version="1.0" encoding="UTF-8" ?>
<data logdt="2016-02-19 00:05:00">
 <device id="1" channel="1">
 <node name="Comm_Status" value="1" unit=""></node>
 <node name="voltA" value="220.1" unit="V"></node>
 <node name="voltB" value="220.1" unit="V"></node>
 <node name="voltC" value="220.1" unit="V"></node>
 <node name="currA" value="5.01" unit="A"></node>
 <node name="currB" value="5.01" unit="A"></node>
 <node name="currC" value="5.01" unit="A"></node>

 </device>
 <device id="2" channel="1">

 </device>

</data>

	<pre>{ "id": "2", "channel": "1", "node": [.....] }, }</pre>			
调用举例	http://192. 168. 13. 64:80/realdata/<gateway_sn>			
参数说明				
参数名称	类型	长度	必填	说明
logdt	时间		必填	
device	节点			注意：channel+id 可以作为同一网关下设备的唯一标志
channel	String	10	必填	通道号
id	String	10	必填	设备通讯地址号
name	String	128	必填	采集点名称，（Comm_Status 是通讯状态点， 0 代表无法通讯，1 通讯正常， -1 通讯有异常报文）
value	String	128	必填	采集点数值
unit	String	32		采集点单位
返回数据				
参数名称	类型	长度	必填	说明
提交申请， 返回值 http code 为 201 ， 表示创建成功。http code 406, 表示创建信息失败。http code 426(Upgrade Required)表示创建成功，并且有下发文件的请求。				

4.8 上传实时告警接口

4.8.1 网关上传告警接口

4.8.1.1 接口定义

接口编号	0006	接口名称	/alarmdata/<gateway_sn>
接口描述	网关告警上传接口		

调用方式	POST			
调用说明	网关告警上传。			
Header 参数	Content-Type:application/json		Content-Type:application/xml	
Body 参数	<pre>{ "node": [{ "logdt": "2016-02-19 00:05:00", "channel": "1", "device": "1", "name": "voltA", "value": "220.1", "threshold_value": "218.4", "unit": "V", "type": "1", "flag": "1" }, ] }</pre>		<pre><?xml version="1.0" encoding="UTF-8" ?> <data> <node logdt="2016-02-19 00:05:00" channel="1" device="1" name="voltA" value="220.1" unit="V" threshold_value="220.1" type="1" flag="1"> </node> </data></pre>	
调用举例	<a href="http://192.168.13.64:80/realdata/<gateway_sn>">http://192.168.13.64:80/realdata/<gateway_sn>			
参数说明				
参数名称	类型	长度	必填	说明
node	节点			
channel	Number	10	必填	通道号
device	Number	5	必填	设备的地址号
logdt	String	20	必填	告警发生时间
threadhold_value	Number	20		设置的阈值
type	Number	2	必填	0:越上限 1:越下限 2:异常
flag	Number	2	必填	0:恢复 1:发生
返回数据				
参数名称	类型	长度	必填	说明
提交申请， 返回值 http code 为 201 ，表示创建成功。http code 406,表示创建信息失败。				

4.9 回召接口

4.9.1 查询网关回召接口

4.9.1.1 接口定义

接口编号	0007	接口名称	/calldata/<gateway_sn>	
接口描述	网关回召查询接口			
调用方式	GET 查询本网关需要回召的数据			
调用说明	网关回召查询			
Header 参数	Content-Type:application/json		Content-Type:application/xml	
Body 参数				
调用举例	<a href="http://192.168.13.64:80/calldata/<gateway_sn>">http://192.168.13.64:80/calldata/<gateway_sn>			
返回数据				
返回参数	<pre>{ "node": [{ "startdt": "2015-12-31 12:00:23", "enddt": "2016-01-01 10:12:31"}, { "startdt": "2016-02-31 12:00:23", "enddt": "2016-03-01 10:12:31"},] }</pre>		<pre><?xml version="1.0" encoding="UTF-8" ?> <data> <node startdt="2016-02-19 00:05:00" enddt="2016-02-19 01:10:00"/> <node startdt="2016-03-19 00:05:00" enddt="2016-03-19 01:10:00"/> </data></pre>	
参数说明				
参数名称	类型	长度	必填	说明
startdt	String	20		回召的起始时间
enddt	String	20		回召的结束时间
返回值 http code 为 200 ， 表示获取成功。http code 404, 表示没有该网关信息。				

4.9.2 删除网关回召接口

4.9.2.1 接口定义

接口编号	0008	接口名称	/calldata/<gateway_sn>
------	------	------	------------------------

接口描述	网关回召删除接口			
调用方式	DELETE 删除本网关需要回召的时间段			
调用说明	网关回召查询			
Header 参数	Content-Type:application/json		Content-Type:application/xml	
Body 参数	<pre>{ "node":[{"startdt":"2015-12-31 12:00:23", "enddt":"2016-01-01 10:12:31"}, {"startdt":"2016-02-31 12:00:23", "enddt":"2016-03-01 10:12:31"},]}</pre>		<pre><?xml version="1.0" encoding="UTF-8" ?> <data> <node startdt="2016-02-19 00:05:00" enddt="2016-02-19 01:10:00"/> <node startdt="2016-03-19 00:05:00" enddt="2016-03-19 01:10:00"/> </data></pre>	
调用举例	<a href="http://192.168.13.64:80/calldata/<gateway_sn>">http://192.168.13.64:80/calldata/<gateway_sn>			
返回数据				
参数说明				
参数名称	类型	长度	必填	说明
startdt	String	20		回召的起始时间
enddt	String	20		回召的结束时间
返回值 http code 为 204 ， 表示删除成功。http code 406, 表示删除失败。				

4.10 下发文件接口

4.10.1 下发接口

4.10.1.1 接口定义

接口编号	0009	接口名称	/downfile/<gateway_sn>
接口描述	网关下发接口接口		
调用方式	GET 查询本网关需要下载的数据		
调用说明	查询本网关需要下载的数据。		
Header 参数	Content-Type:application/json		Content-Type:application/xml
Body 参数	<pre>{ "node": [{ "file": "/down/update.pkg", "md5": "9798a99f0add45a8f6c4d31e5e88281d", },] }</pre>		<pre><?xml version="1.0" encoding="UTF-8" ?> <data> <node file="/down/update.pkg", md5="9798a99f0add45a8f6c4d31e5e88281d", ></pre>

	<pre>"type": "0", "path": "/tmp/uploads"}, {"file": "/down/config.conf", "md5": "9798a99f0add45a8f6c4d31e5e88281d", "type": "0", "path": "/tmp/uploads"}] }</pre>	<pre>type="0", path="/tmp/uploads"/> <node file="/down/config.conf", md5="9798a99f0add45a8f6c4d31e5e88281d", type="0", path="/tmp/uploads"/> </data></pre>		
调用举例	<a href="http://192.168.13.64:80/downloadfile/<gateway_sn>">http://192.168.13.64:80/downloadfile/<gateway_sn>			
参数说明				
参数名称	类型	长度	必填	说明
file	String	200		下载文件的具体路径
md5	String	20		下载文件的 md5
type	int	4		文件类型: 0-固件包 update.pkg 1-配置包 config.conf 9-其它类型
path	String	50		网关中存放的位置，主要当 type 等于 9 的时候有意义
返回数据				
参数名称	类型	长度	必填	说明
返回值 http code 为 200 ，表示获取成功。http code 404,表示没有该网关信息。				

4.10.2 下发更改状态接口

4.10.2.1 接口定义

接口编号	0010	接口名称	/downloadfile/<gateway_sn>
接口描述	网关下发接口接口		
调用方式	PUT 修改本网关需要下载的数据记录状态 状态 0-未处理 1-已收到 2-处理完成		
调用说明	修改本网关需要下载的数据记录状态		
Header 参数	Content-Type:application/json		Content-Type:application/xml
Body 参数	<pre> { "node": [{ "md5": "9798a99f0add45a8f6c4d31e5e88281d", "status": "1" }] } </pre>		<pre> <?xml version="1.0" encoding="UTF-8" ?> <data> <node md5="9798a99f0add45a8f6c4d31e5e88281d", status="1"/> </data> </pre>

调用举例	<a href="http://192.168.13.64:80/downloadfile/<gateway_sn>">http://192.168.13.64:80/downloadfile/<gateway_sn>			
参数说明				
参数名称	类型	长度	必填	说明
md5	String	20		下载文件的 md5
status	int	4		状态 0-未处理 1-已收到 2-处理完成
返回数据				
参数名称	类型	长度	必填	说明
返回值 http code 为 201 ，表示成功。http code 406,表示失败。				

4.11 上位机开发说明

4.11.1 实现接口说明

接口名称	实现	说明
realdata	必须实现	网关实时数据上传接口，用来接收实时数据
alarmdata	必须实现	网关告警上传接口，如没有特殊配置的话，该接口不会返回数据，但必须要实现
gateway	必须实现	网关创建接口，网关一开始运行的时候会调用一次（重启网关之后也会调用一次），必须要实现该接口
Gateway/{gateway_sn}	必须实现	查看网关接口，网关一开始运行的时候会调用一次，必须要实现该接口
device/{gateway_sn} (Post 请求)	必须实现	创建设备接口，必须要实现该接口
device/{gateway_sn} (Get 请求)	必须实现	查看设备接口，必须要实现该接口
备注：如继续正常运行，上面接口必须返回 HttpStatusCode.Created (201) Code		

4.11.2 C# 代码示例

```
#region Gate实时数据上传处理
```

```
/// <summary>
/// Gate 3G/4G网络版本
/// 说明：POST请求
/// WebInvoke请求方式有POST、PUT、DELETE等，所以需要明确指定Method是哪种请求
/// 的，这里我们设置POST请求。
/// UriTemplate(URL Routing)input必须要方法的参数名必须一致不区分大小写)
/// RequestFormat规定客户端必须是什么数据格式请求的（Json或者XML），不设置默认为XML
/// ResponseFormat规定服务端返回给客户端是以是什么数据格返回的（Json或者XML）
[OperationContract]
[WebInvoke(Method = "POST", UriTemplate =
"GateRealData/realdata/{gateway_sn}", BodyStyle = WebMessageBodyStyle.Bare,
RequestFormat = WebMessageFormat.Json, ResponseFormat = WebMessageFormat.Json)]
void GateRealDataReceive(RealDataReceiveItem realdata, string gateway_sn);

/// <summary>
/// 告警数据上传接口
/// </summary>
/// <param name="alarmdata"></param>
/// <param name="gateway_sn"></param>
[OperationContract]
[WebInvoke(Method = "POST", UriTemplate =
"GateRealData/alarmdata/{gateway_sn}", BodyStyle = WebMessageBodyStyle.Bare,
RequestFormat = WebMessageFormat.Json, ResponseFormat = WebMessageFormat.Json)]
void GateAlarmData(GateAlarmDataBody alarmdata, string gateway_sn);

/// <summary>
/// 创建网关接口，一般不会用到，但是必须要有。
/// </summary>
/// <param name="jsonStr"></param>
/// <returns></returns>
[OperationContract]
[WebInvoke(Method = "POST", UriTemplate = "GateRealData/gateway", BodyStyle
= WebMessageBodyStyle.Bare, RequestFormat = WebMessageFormat.Json, ResponseFormat =
WebMessageFormat.Json)]
void CreateGateway(GateWayInfoItem gateWayInfo);

/// <summary>
/// 查看网关接口，一般不会用到，但是必须要有。
/// </summary>
/// <param name="jsonStr"></param>
/// <param name="gateway_sn"></param>
/// <returns></returns>
[OperationContract]
[WebInvoke(Method = "GET", UriTemplate = "GateRealData/gateway/{gateway_sn}",
```

```

BodyStyle = WebMessageBodyStyle.Bare, RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json)]
    void GetGateway(string gateway_sn);

    /// <summary>
    /// 创建设备接口，一般不会用到，但是必须要有。
    /// </summary>
    /// <param name="jsonStr"></param>
    /// <returns></returns>
    [OperationContract]
    [WebInvoke(Method = "POST", UriTemplate = "GateRealData/device/{gateway_sn}",
BodyStyle = WebMessageBodyStyle.Bare, RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json)]
    void CreateDevice(GateWayDeviceInfo deviceInfo, string gateway_sn);

    /// <summary>
    /// 查看设备接口，一般不会用到，但是必须要有。
    /// </summary>
    /// <param name="jsonStr"></param>
    /// <param name="gateway_sn"></param>
    /// <returns></returns>
    [OperationContract]
    [WebInvoke(Method = "GET", UriTemplate = "GateRealData/device/{gateway_sn}",
BodyStyle = WebMessageBodyStyle.Bare, RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json)]
    void GetDevice(string gateway_sn);
#endregion

/// <summary>
/// Gate/XGate 3G/4G 实时数据接收模型
/// </summary>
[DataContract]
[Serializable]
public class RealDataReceiveItem
{
    /// <summary>
    /// 时间
    /// </summary>
    [DataMember]
    public string logdt { get; set; }

    /// <summary>
    /// 具体内容
    /// </summary>
    [DataMember]

```



```
        public List<RealDataBody> device { get; set; } = new List<RealDataBody>();
    }
    [DataContract]
    [Serializable]
    public class RealDataBody
    {
        /// <summary>
        /// Gate 通道
        /// </summary>
        [DataMember]
        public int channel { get; set; }
        /// <summary>
        /// Gate通道下面的设备，一般情况下为1，因为我们可能只会用到一个
        /// </summary>
        [DataMember]
        public int id { get; set; }
        /// <summary>
        /// 具体值
        /// </summary>
        [DataMember]
        public List<XGateUpload> node { get; set; } = new List<XGateUpload>();
    }

    /// <summary>
    /// XGate 3G/4G 上传实体
    /// </summary>
    [DataContract]
    [Serializable]
    public class XGateUpload
    {
        /// <summary>
        /// 点地址
        /// </summary>
        [DataMember]
        public string name { get; set; }
        /// <summary>
        /// 单位
        /// </summary>
        [DataMember]
        public string unit { get; set; }
        /// <summary>
        /// 值
        /// </summary>
```

```
[DataMember]
public decimal value { get; set; }
}
```

4.12 服务器接收数据流解析说明

4.12.1 Name 属性名定义

Name 属性名	描述
组一单体实时数据（电压）	
8	组 1_电池单体_1_单体电压
10	组 1_电池单体_2_单体电压
12	组 1_电池单体_3_单体电压
14	组 1_电池单体_4_单体电压
...	...
...	...
244	组 1_电池单体_119_单体电压
246	组 1_电池单体_120_单体电压
组一单体实时数据（温度）	
248	组 1_电池单体_1_单体温度
250	组 1_电池单体_2_单体温度
252	组 1_电池单体_3_单体温度
254	组 1_电池单体_4_单体温度
...	...
...	...
484	组 1_电池单体_119_单体温度
486	组 1_电池单体_120_单体温度
组一单体实时数据（内阻）	

488	组 1_电池单体_1_单体内阻
490	组 1_电池单体_2_单体内阻
492	组 1_电池单体_3_单体内阻
494	组 1_电池单体_4_单体内阻
...	...
...	...
724	组 1_电池单体 119_单体内阻
726	组 1_电池单体 120_单体内阻
组一单体实时数据 (SOC)	
728	组 1_电池单体_1_单体 soc
730	组 1_电池单体_2_单体 soc
732	组 1_电池单体_3_单体 soc
734	组 1_电池单体_4_单体 soc
...	...
...	...
964	组 1_电池单体 119_单体 soc
966	组 1_电池单体_120_单体 soc
组一单体实时数据 (SOH)	
968	组 1_电池单体_1_单体 soh
970	组 1_电池单体_2_单体 soh
972	组 1_电池单体_3_单体 soh
974	组 1_电池单体_4_单体 soh
...	...
...	...
1204	组 1_电池单体 119_单体 soh

1206	组 1_电池单体_120_单体 soh
------	---------------------

Name 属性名	描述
组二单体实时数据（电压）	
1208	组 2_电池单体_1_单体电压
1210	组 2_电池单体_2_单体电压
1212	组 2_电池单体_3_单体电压
1214	组 2_电池单体_4_单体电压
...	...
...	...
1444	组 2_电池单体_119_单体电压
1446	组 2_电池单体_120_单体电压
组二单体实时数据（温度）	
1448	组 2_电池单体_1_单体温度
1450	组 2_电池单体_2_单体温度
1452	组 2_电池单体_3_单体温度
1454	组 2_电池单体_4_单体温度
...	...
...	...
1684	组 2_电池单体_119_单体温度
1686	组 2_电池单体_120_单体温度
组二单体实时数据（内阻）	
1688	组 2_电池单体_1_单体内阻
1690	组 2_电池单体_2_单体内阻
1692	组 2_电池单体_3_单体内阻

1694	组 2_电池单体_4_单体内阻
...	...
...	...
1924	组 2_电池单体_119_单体内阻
1926	组 2_电池单体_120_单体内阻
组二单体实时数据 (SOC)	
1928	组 2_电池单体_1_单体 soc
1930	组 2_电池单体_2_单体 soc
1932	组 2_电池单体_3_单体 soc
1934	组 2_电池单体_4_单体 soc
...	...
...	...
2164	组 2_电池单体_119_单体 soc
2166	组 2_电池单体_120_单体 soc
组二单体实时数据 (SOH)	
2168	组 2_电池单体_1_soh
2170	组 2_电池单体_2_soh
2172	组 2_电池单体_3_soh
2174	组 2_电池单体_4_soh
...	...
...	...
2404	组 2_电池单体_119_soh
2406	组 2_电池单体_120_soh

Name 属性名	描述
组三单体实时数据 (电压)	

2408	组 3_电池单体_1_单体电压
2410	组 3_电池单体_2_单体电压
2412	组 3_电池单体_3_单体电压
2414	组 3_电池单体_4_单体电压
...	...
...	...
2644	组 3_电池单体_119_单体电压
2646	组 3_电池单体_120_单体电压
组三单体实时数据（温度）	
2648	组 3_电池单体_1_单体温度
2650	组 3_电池单体_2_单体温度
2652	组 3_电池单体_3_单体温度
2654	组 3_电池单体_4_单体温度
...	...
...	...
2884	组 3_电池单体_119_单体温度
2886	组 3_电池单体_120_单体温度
组三单体实时数据（内阻）	
2888	组 3_电池单体_1_单体内阻
2890	组 3_电池单体_2_单体内阻
2892	组 3_电池单体_3_单体内阻
2894	组 3_电池单体_4_单体内阻
...	...
...	...
3124	组 3_电池单体_119_单体内阻

3126	组 3_电池单体_120_单体内阻
组三单体实时数据（SOC）	
3128	组 3_电池单体_1_单体 soc
3130	组 3_电池单体_2_单体 soc
3132	组 3_电池单体_3_单体 soc
3134	组 3_电池单体_4_单体 soc
...	...
...	...
3364	组 3_电池单体_119_单体 soc
3366	组 3_电池单体_120_单体 soc
组三单体实时数据（SOH）	
3368	组 3_电池单体_1_单体 soh
3370	组 3_电池单体_2_单体 soh
3372	组 3_电池单体_3_单体 soh
3374	组 3_电池单体_4_单体 soh
...	...
...	...
3604	组 3_电池单体_119_单体 soh
3606	组 3_电池单体_120_单体 soh

Name 属性名	描述
组四单体实时数据（电压）	
3608	组 4_电池单体_1_单体电压
3610	组 4_电池单体_2_单体电压
3612	组 4_电池单体_3_单体电压

3614	组 4_电池单体_4_单体电压
...	...
...	...
3844	组 4_电池单体_119_单体电压
3846	组 4_电池单体_120_单体电压
组四单体实时数据（温度）	
3848	组 4_电池单体_1_单体温度
3850	组 4_电池单体_2_单体温度
3852	组 4_电池单体_3_单体温度
3854	组 4_电池单体_4_单体温度
...	...
...	...
4084	组 4_电池单体_119_单体温度
4086	组 4_电池单体_120_单体温度
组四单体实时数据（内阻）	
4088	组 4_电池单体_1_单体内阻
4090	组 4_电池单体_2_单体内阻
4092	组 4_电池单体_3_单体内阻
4094	组 4_电池单体_4_单体内阻
...	...
...	...
4324	组 4_电池单体_119_单体内阻
4326	组 4_电池单体_120_单体内阻
组四单体实时数据（SOC）	
4328	组 4_电池单体_1_单体 soc
4330	组 4_电池单体_2_单体 soc

4332	组 4_电池单体_3_单体 soc
4334	组 4_电池单体_4_单体 soc
...	...
...	...
4564	组 4_电池单体_119_单体 soc
4566	组 4_电池单体_120_单体 soc
组四单体实时数据（SOH）	
4568	组 4_电池单体_1_单体 soh
4570	组 4_电池单体_2_单体 soh
4572	组 4_电池单体_3_单体 soh
4574	组 4_电池单体_4_单体 soh
...	...
...	...
4804	组 4_电池单体_119_单体 soh
4806	组 4_电池单体_120_单体 soh

Name 属性名	描述
组体实时数据	
4808	电池组 1_组 电压
4810	电池组 2_组 电压
4812	电池组 3_组 电压
4814	电池组 4_组 电压
4816	电池组 1_组 电流
4818	电池组 2_组 电流
4820	电池组 3_组 电流

4822	电池组 4_组 电流
4824	电池组 1_SOC
4826	电池组 2_SOC
4829	电池组 3_SOC
4830	电池组 4_SOC
4832	组 1_电池组_均衡度
4834	组 2_电池组_均衡度
4836	组 3_电池组_均衡度
4838	组 4_电池组_均衡度
4840	组 1_电池组_电池组状态
4842	组 2_电池组_电池组状态
4844	组 3_电池组_电池组状态
4846	组 4_电池组_电池组状态

Name 属性名		描述
模拟量状态		
5864	一路 DI 状态	1 表示接通，0 表示断开。
5866	二路 DI 状态	1 表示接通，0 表示断开。
5868	三路 DI 状态	1 表示接通，0 表示断开。
5870	四路 DI 状态	1 表示接通，0 表示断开。
5874	环境温度	
5876	环境湿度	

Name 属性名		描述
电池连接状态		
5878	组 1_电池单体_1_电池连接状态	Bit0:通讯状态，1 表示正常，0 表示中断。
5880	组 1_电池单体_2_电池连接状态	Bit0:通讯状态，1 表示正常，0 表示中断。
5882	组 1_电池单体_3_电池连接状态	Bit0:通讯状态，1 表示正常，0 表示中断。
5884	组 1_电池单体_4_电池连接状态	Bit0:通讯状态，1 表示正常，0 表示中断。
...
...
6834	组 4_电池单体_119_电池连接状态	Bit0:通讯状态，1 表示正常，0 表示中断。
6836	组 4_电池单体_120_电池连接状态	Bit0:通讯状态，1 表示正常，0 表示中断。

4.12.2 接收数据流解析实例

以 json 数据流为例，Web Service 端接收 Json 数据流如下图 1，各主要字段说明：

channel: Gate 通道号，如 1 表示 Gate 通道口 1

id: 采集设备地址号，如 1 表示组采集器 PBAT600 地址为 1；

name: 采集数据属性名，参照章节 4.12.1 中 Name 属性名定义，如红色方框中 5878 表示组 1_电池单体_1_电池连接状态

unit: 采集数据单位符号，如“#”表示无具体单位，“mΩ”表示内阻单位为毫欧

value: 采集数据对应值，如 5878 对应值为 0 表示组 1_电池单体_1_电池连接状态未连接（中断）

```
2019121521log.txt
C: > Users > user > Desktop > 2019121521log.txt
1  -- -- -- -- --2019 / 12 / 15 21: 00: 55-- -- -- -- --
2  Gate 2019 / 12 / 15 21: 00: 55: GW1501011111 {
3      "logdt": "2019-12-15 21:00:00",
4      "device": [{
5          "channel": 1,
6          "id": 1,
7          "node": [{
8              "name": "5878",
9              "unit": "#",
10             "value": 0.000
11         }, {
12             "name": "5880",
13             "unit": "#",
14             "value": 0.000
15         }, {
16             "name": "5882",
17             "unit": "#",
18             "value": 0.000
19         }, {
20             "name": "5884",
21             "unit": "#",
22             "value": 0.000
23         }, {
24             "name": "5886",
25             "unit": "#",
26             "value": 0.000
27         }, {
28             "name": "5888",
29             "unit": "#",
30             "value": 0.000
31         }, {
32             "name": "5890",
33             "unit": "#",
34             "value": 0.000
35         }, {
36             "name": "5892",
37             "unit": "#",
38             "value": 0.000
39         }, {
40             "name": "5894",
41             "unit": "#",
42             "value": 0.000
43         }, {
44             "name": "5896",
45             "unit": "#",
46             "value": 0.000
47         }, {
48             "name": "5898",
49             "unit": "#",
50             "value": 0.000
51         }
52     ]
53 }
```

图 1

5.1 附录

5.1.1 返回 http code 状态定义

RESTFUL 框架服务器常见返回状态码和提示信息，常见的有以下（方括号中是该状态码对应的 HTTP 动词）。

- 200 OK - [GET]: 服务器成功返回用户请求的数据，该操作是幂等的（Idempotent）。
- 201 CREATED - [POST/PUT/PATCH]: 用户新建或修改数据成功。
- 202 Accepted - [*]: 表示一个请求已经进入后台排队（异步任务）
- 204 NO CONTENT - [DELETE]: 用户删除数据成功。
- 400 INVALID REQUEST - [POST/PUT/PATCH]: 用户发出的请求有错误，服务器没有进行新建或修改数据的操作，该操作是幂等的。
- 401 Unauthorized - [*]: 表示用户没有权限（令牌、用户名、密码错误）。
- 403 Forbidden - [*] 表示用户得到授权（与 401 错误相对），但是访问是被禁止的。
- 404 NOT FOUND - [*]: 用户发出的请求针对的是不存在的记录，服务器没有进行操作，该操作是幂等的。
- 406 Not Acceptable - [GET]: 用户请求的格式不可得（比如用户请求 JSON 格式，但是只有 XML 格式）。
- 410 Gone -[GET]: 用户请求的资源被永久删除，且不会再得到的。
- 422 Unprocesable entity - [POST/PUT/PATCH] 当创建一个对象时，发生一个验证错误。
- 500 INTERNAL SERVER ERROR - [*]: 服务器发生错误，用户将无法判断发出的请求是否成功。