

# GNSS 与物联网平台数据对接协议

2021 年 10 月

# 目录

<b>1. 范围</b>	<b>1</b>
<b>2. Rest 接口</b>	<b>1</b>
2.1. 接口概述	1
2.1.1. URI 定义	1
2.1.2. 公共请求头	1
2.1.3. 公共响应头	1
2.1.4. http 响应码	2
2.1.5. DeviceData 格式定义	2
2.2. 服务注册接口[regist]	3
2.2.1. 调用方向	3
2.2.2. 方法和 uri	3
2.2.3. 参数说明和示例	3
2.2.4. 响应码	4
2.2.5. 响应数据说明	4
2.3. 数据上报接口[report]	4
2.3.1. 调用方向	4
2.3.2. 方法和 uri	4
2.3.3. 参数说明和示例	4
2.3.4. 响应码	5

2.3.5.	响应数据说明	5
2.4.	<i>查询接口[query]</i>	6
2.4.1.	调用方向	6
2.4.2.	方法和 uri	6
2.4.3.	参数说明和示例	6
2.4.4.	响应码	7
2.4.5.	响应数据说明	7
2.5.	<i>异步查询应答接口[queryAck]</i>	8
2.5.1.	调用方向	8
2.5.2.	方法和 uri	9
2.5.3.	参数说明和示例	9
2.5.4.	响应码	10
2.5.5.	响应数据说明	10
2.6.	<i>配置接口[config]</i>	11
2.6.1.	调用方向	11
2.6.2.	方法和 uri	11
2.6.3.	参数说明和示例	11
2.6.4.	响应码	12
2.6.5.	响应数据说明	12
2.7.	<i>异步配置应答接口[configAck]</i>	13
2.7.1.	调用方向	13
2.7.2.	方法和 uri	13

2.7.3.	参数说明和示例	13
2.7.4.	响应码	15
2.7.5.	响应数据说明	15
2.8.	<i>控制接口[control]</i>	15
2.8.1.	调用方向	15
2.8.2.	方法和 uri	15
2.8.3.	参数说明和示例	15
2.8.4.	响应码	16
2.8.5.	响应数据说明	16
2.9.	<i>异步控制应答接口[controlAck]</i>	17
2.9.1.	调用方向	17
2.9.2.	方法和 uri	18
2.9.3.	参数说明和示例	18
2.9.4.	响应码	19
2.9.5.	响应数据说明	19

## 1. 范围

本文档定义了物联网平台通过服务对接方式对三方系统开放的 rest 接口

## 2. Rest 接口

### 2.1. 接口概述

#### 2.1.1. URI 定义

对象	URI	标记
物联网平台	/iotplatform	<platform_basic_uri>
上行消息	<platform_basic_uri>/up	<platform_up_uri>
三方应用注册到物联网平台的下行 url	--	<regist-downurl>

#### 2.1.2. 公共请求头

名称	属性	是否必选	描述
Content-Type	String	是	表示相应的实体的 MIME 类型, 本系统为 json 格式, 语法: application/json;charset=UTF-8
X-Auth	String	是	鉴权用户名
X-Token	String	是	鉴权密码
msgSeq	String	是	指令序列标识,uuid (36)
msgType	String	是	标识该报文所属消息类型, 如: report、ctrl、query、config 等
resultCode	String	否	异步指令执行结果: 0, 成功 1, 失败

#### 2.1.3. 公共响应头

名称	描述	是否必须	描述
----	----	------	----

Content-Type		必选	表示相应的实体的 MIME 类型，本系统为 json 格式，语法：application/json;charset=UTF-8
resultCode	String	否	指令执行结果： 0，成功 1，失败 2，执行中（异步执行，执行结果应答消息会通过上行消息发送给物联网平台）

#### 2.1.4. http 响应码

编码	含义	描述
200	OK, 成功	操作成功
204	空内容	删除成功
400	Bad Request, 错误的请求	请求内容缺失或无效的请求内容
401	Unauthorized, 无权限	没有权限调用该接口
403	Forbidden, 禁止操作	用户不存在, 不允许执行该操作
404	NOT FOUND, 无此对象	要操作的对象不存在
500	Internal Server Error, 服务器内部错误	服务器遇到了一个未曾预料的状态, 导致了它无法完成对请求的处理。
503	服务器超载	服务器总请求数超过流控限制。
600	操作出现失败	批量操作中, 部分操作执行失败

#### 2.1.5. DeviceData 格式定义

响应体为 json 格式用 utf-8 编码

deviceData 格式:

项目	属性	说明
deviceCode	String	设备标识
timestamp	String	时间戳, 需按时间顺序发送数据
metric1	object	厂商自定义采集指标, 要求按采集设备定义该设备可以采集的数据
metric2	object	
...	object	

metricn	object	
---------	--------	--

## 2.2. 服务注册接口[regist]

### 2.2.1. 调用方向

三方系统作为客户端通过 rest 接口注册下行服务地址到物联网平台

### 2.2.2. 方法和 uri

方法	URI	描述
Post	< platform_basic_uri>/regist	注册下行服务接口

### 2.2.3. 参数说明和示例

Body 参数:

参数名	类型	说明
url	String	下行服务地址

示例:

请求头:

名称	示例
Content-Type	application/json;charset=UTF-8
X-Auth	User
X-Token	Pass
msgSeq	c1628f43-16a6-4d87-bb72-21b2961a7b10
msgType	regist

Body:

<pre>{   "url": <a href="http://down.url.com:8080/downmsg">http://down.url.com:8080/downmsg</a> }</pre>
---------------------------------------------------------------------------------------------------------

## 2.2.4. 响应码

参照 2.1.4http 响应码

## 2.2.5. 响应数据说明

响应示例:

响应头:

名称	示例
Content-Type	application/json;charset=UTF-8

Body:

无
---

## 2.3. 数据上报接口[report]

### 2.3.1. 调用方向

三方系统作为客户端通过 rest 接口发送数据上报(report)物联网平台。

### 2.3.2. 方法和 uri

方法	URI	描述
Post	<platform_up_uri>	上行消息

### 2.3.3. 参数说明和示例

Body 格式:

项目	属性	说明
Body	List(DeviceData)	设备数据数组

DeviceData 具体格式参见 2.1.5

请求示例:

请求头:

名称	示例
Content-Type	application/json;charset=UTF-8
X-Auth	User
X-Token	Pass
msgSeq	c1628f43-16a6-4d87-bb72-21b2961a7b10
msgType	report

Body:

<pre>[   {     "deviceCode": "abc001",     "timestamp": "2021-09-30 14:52:51",     "ph": 7,     "totalflow": 12345678   },   {     "deviceCode": "abc002",     "timestamp": "2021-09-30 14:52:51",     "ph": 7,     "totalflow": 12345678   } ]</pre>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### 2.3.4. 响应码

http 响应码参照 2.1.4http 响应码

### 2.3.5. 响应数据说明

应答示例:

应答头:

名称	示例
Content-Type	application/json;charset=UTF-8

Body:

无
---

## 2.4. 查询接口[query]

### 2.4.1. 调用方向

物联网平台作为客户端通过 rest 接口调用注册的服务地址，发送查询指令到三方系统。

### 2.4.2. 方法和 uri

方法	URI	描述
Post	<regist-downurl>	三方系统注册的 url

### 2.4.3. 参数说明和示例

Message Body 格式:

项目	属性	说明
Body	List(DeviceData)	设备数据数组

DeviceData 具体格式参见 2.1.5

示例:

请求头:

名称	示例
Content-Type	application/json;charset=UTF-8
X-Auth	User
X-Token	Pass
msgSeq	c1628f43-16a6-4d87-bb72-21b2961a7b10
msgType	query

Body:

```
[
  {
    "deviceCode": "abc001",
    "timestamp": "2021-09-30 14:52:51",
    "port": 0,
    "ip": ""
  },
  {
    "deviceCode": "abc002",
    "timestamp": "2021-09-30 14:52:51",
```

```
        "port":0,  
        "ip":""  
    }  
]
```

### 2.4.4. 响应码

参照 2.1.4http 响应码

### 2.4.5. 响应数据说明

响应 body:

DeviceData 具体格式参见 2.1.5

resultCode =0

项目	属性	说明
Body	List(DeviceData)	设备数据数组

resultCode =1

项目	属性	说明
result	List(DeviceData)	设备数据数组，查询成功设备
errorMsg	String	错误信息

resultCode =2

无 body，等异步查询应答返回查询结果

应答示例:

应答头:

名称	示例
Content-Type	application/json;charset=UTF-8
resultCode	0

Body:

resultCode =0

```
[  
  {  
    ...  
  }  
]
```

```
    "deviceCode": "abc001",
    "timestamp": "2021-09-30 14:52:51",
    "port": 2104,
    "ip": "123.4.56.78"
  },
  {
    "deviceCode": "abc002",
    "timestamp": "2021-09-30 14:52:51",
    "port": 2104,
    "ip": "123.4.56.78"
  }
]
```

resultCode =1

```
{
  "errorMsg": "error",
  "result": [{
    "deviceCode": "abc001",
    "timestamp": "2021-09-30 14:52:51",
    "port": 2104,
    "ip": "123.4.56.78"
  },
  {
    "deviceCode": "abc002",
    "timestamp": "2021-09-30 14:52:51",
    "port": 2104,
    "ip": "123.4.56.78"
  }
]
}
```

resultCode =2

```
无
```

## 2.5. 异步查询应答接口[queryAck]

### 2.5.1. 调用方向

三方系统作为客户端通过 rest 接口发送查询应答数据到物联网平台。

## 2.5.2. 方法和 uri

方法	URI	描述
Post	<platform_up_uri>	上行消息

## 2.5.3. 参数说明和示例

Body:

DeviceData 具体格式参见 2.1.5

resultCode =0

项目	属性	说明
Body	List(DeviceData)	设备数据数组

resultCode =1

项目	属性	说明
result	List(DeviceData)	设备数据数组，查询成功设备
errorMsg	String	错误信息

请求示例:

请求头:

名称	示例
Content-Type	application/json;charset=UTF-8
X-Auth	User
X-Token	Pass
msgSeq	c1628f43-16a6-4d87-bb72-21b2961a7b10
msgType	queryAck
resultCode	0

Body:

resultCode =0

```
[
  {
    "deviceCode": "abc001",
    "timestamp": "2021-09-30 14:52:51",
    "port": 2104,
    "ip": "123.4.56.78"
  },
  {
    "deviceCode": "abc002",
```

```
        "timestamp": "2021-09-30 14:52:51",
        "port": 2104,
        "ip": "123.4.56.78"
    }
]
```

resultCode =1

```
{
  "errorMsg": "error",
  "result": [{
    "deviceCode": "abc001",
    "timestamp": "2021-09-30 14:52:51",
    "port": 2104,
    "ip": "123.4.56.78"
  },
  {
    "deviceCode": "abc002",
    "timestamp": "2021-09-30 14:52:51",
    "port": 2104,
    "ip": "123.4.56.78"
  }
]
}
```

#### 2.5.4. 响应码

http 响应码参照 2.1.4http 响应码

#### 2.5.5. 响应数据说明

应答示例:

应答头:

名称	示例
Content-Type	application/json;charset=UTF-8

Body:

无

## 2.6. 配置接口[config]

### 2.6.1. 调用方向

物联网平台作为客户端通过 rest 接口调用注册的服务地址，发送配置指令到三方系统。

### 2.6.2. 方法和 uri

方法	URI	描述
Post	<regist-downurl>	三方系统注册的 url

### 2.6.3. 参数说明和示例

Message Body 格式:

项目	属性	说明
Body	List(DeviceData)	设备数据数组

DeviceData 具体格式参见 2.1.5

示例:

请求头:

名称	示例
Content-Type	application/json;charset=UTF-8
X-Auth	User
X-Token	Pass
msgSeq	c1628f43-16a6-4d87-bb72-21b2961a7b10
msgType	config

Body:

```
[
  {
    "deviceCode": "abc001",
    "timestamp": "2021-09-30 14:52:51",
    "port": 2014,
    "ip": "123.45.6.78"
  },
  {
```

```
[
  {
    "deviceCode": "abc002",
    "timestamp": "2021-09-30 14:52:51",
    "port": 2014,
    "ip": "123.45.6.78"
  }
]
```

### 2.6.4. 响应码

参照 2.1.4http 响应码

### 2.6.5. 响应数据说明

响应 body:

DeviceData 具体格式参见 2.1.5

resultCode =0

无

resultCode =1

项目	属性	说明
result	List(DeviceData)	设备数据数组，配置成功设备
errorMsg	String	错误信息

resultCode =2

无 body，等异步配置应答返回查询结果

应答示例:

应答头:

名称	示例
Content-Type	application/json;charset=UTF-8
resultCode	0

Body:

resultCode =0

无

resultCode =1

```
{
  "errorMsg": "error",
  "result": [{
    "deviceCode": "abc001",
    "timestamp": "2021-09-30 14:52:51",
    "port": 2104,
    "ip": "123.4.56.78"
  },
  {
    "deviceCode": "abc002",
    "timestamp": "2021-09-30 14:52:51",
    "port": 2104,
    "ip": "123.4.56.78"
  }
]
}
```

resultCode =2

无

## 2.7. 异步配置应答接口[configAck]

### 2.7.1. 调用方向

三方系统作为客户端通过 rest 接口发送配置应答数据到物联网平台。

### 2.7.2. 方法和 uri

方法	URI	描述
Post	<platform_up_uri>	上行消息

### 2.7.3. 参数说明和示例

Body:

DeviceData 具体格式参见 2.1.5

resultCode =0

无

resultCode =1

项目	属性	说明
result	List(DeviceData)	设备数据数组，成功配置设备
errorMsg	String	错误信息

请求示例：

请求头：

名称	示例
Content-Type	application/json;charset=UTF-8
X-Auth	User
X-Token	Pass
msgSeq	c1628f43-16a6-4d87-bb72-21b2961a7b10
msgType	configAck
resultCode	0

Body:

resultCode =0

无
---

resultCode =1

<pre>{   "errorMsg": "error",   "result": [{     "deviceCode": "abc001",     "timestamp": "2021-09-30 14:52:51",     "port": 2104,     "ip": "123.4.56.78"   },   {     "deviceCode": "abc002",     "timestamp": "2021-09-30 14:52:51",     "port": 2104,     "ip": "123.4.56.78"   }   ] }</pre>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## 2.7.4. 响应码

http 响应码参照 2.1.4http 响应码

## 2.7.5. 响应数据说明

应答示例:

应答头:

名称	示例
Content-Type	application/json;charset=UTF-8

Body:

无
---

## 2.8. 控制接口[control]

### 2.8.1. 调用方向

物联网平台作为客户端通过 rest 接口调用注册的服务地址，发送控制指令到三方系统。

### 2.8.2. 方法和 uri

方法	URI	描述
Post	<regist-downurl>	三方系统注册的 url

### 2.8.3. 参数说明和示例

Message Body 格式:

项目	属性	说明
Body	List(DeviceData)	设备数据数组

DeviceData 具体格式参见 2.1.5

示例:

请求头:

名称	示例
Content-Type	application/json;charset=UTF-8
X-Auth	User
X-Token	Pass
msgSeq	c1628f43-16a6-4d87-bb72-21b2961a7b10
msgType	control

Body:

```
[
  {
    "deviceCode": "abc001",
    "timestamp": "2021-09-30 14:52:51",
    "valve": 0
  },
  {
    "deviceCode": "abc002",
    "timestamp": "2021-09-30 14:52:51",
    "valve": 1
  }
]
```

## 2.8.4. 响应码

参照 2.1.4http 响应码

## 2.8.5. 响应数据说明

响应 body:

resultCode =0

无

resultCode =1

DeviceData 具体格式参见 2.1.5

项目	属性	说明
result	List(DeviceData)	设备数据数组, 控制成功设备
errorMsg	String	错误信息

resultCode =2

无 body， 等异步控制应答返回控制结果

应答示例：

应答头：

名称	示例
Content-Type	application/json;charset=UTF-8
resultCode	0

Body:

resultCode =0

无

resultCode =1

```
{
  "errorMsg": "error",
  "result": [{
    "deviceCode": "abc001",
    "timestamp": "2021-09-30 14:52:51",
    "valve": 0
  },
  {
    "deviceCode": "abc002",
    "timestamp": "2021-09-30 14:52:51",
    "valve": 1
  }
]
}
```

resultCode =2

无

## 2.9. 异步控制应答接口[controlAck]

### 2.9.1. 调用方向

三方系统作为客户端通过 rest 接口发送控制应答数据到物联网平台。

## 2.9.2. 方法和 uri

方法	URI	描述
Post	<platform_up_uri>	上行消息

## 2.9.3. 参数说明和示例

Body:

resultCode =0

无

resultCode =1

项目	属性	说明
result	List(DeviceData)	设备数据数组，控制成功设备
errorMsg	String	错误信息

请求示例:

请求头:

名称	示例
Content-Type	application/json;charset=UTF-8
X-Auth	User
X-Token	Pass
msgSeq	c1628f43-16a6-4d87-bb72-21b2961a7b10
msgType	controlAck
resultCode	0

Body:

resultCode =0

无

resultCode =1

```
{
  "errorMsg": "error",
  "result": [{
    "deviceCode": "abc001",
    "timestamp": "2021-09-30 14:52:51",
    "valve": 0
  },
  {
    "deviceCode": "abc002",
```

```
        "timestamp": "2021-09-30 14:52:51",
        "valve": 1
    }
]
}
```

#### 2.9.4. 响应码

http 响应码参照 2.1.4http 响应码

#### 2.9.5. 响应数据说明

应答示例:

应答头:

名称	示例
Content-Type	application/json;charset=UTF-8

Body:

无
---