OneNET MQTT能力接入说明书

|  |  |  |  |
| --- | --- | --- | --- |
| **版本号** | **修订日期** | **修订内容** | **说明** |
| V0.1.0 | 2018/12/10 | 草稿 | GuoPei |
| V0.2.0 | 2019/02/19 | 重构 | GuoPei |
| V0.2.1 | 2019/03/07 | 重构 | YuanHao |
| V0.2.2 | 2019/06/19 | 重构 | GuoPei |
| V0.2.3 | 2019/08/12 | 1、简化数据格式，目前只支持type2,3,5三种数据格式 | YuanHao |

目 录

[一、 平台创建MQTT产品 2](#_Toc11830287)

[二、 平台创建MQTT设备 4](#_Toc11830288)

[三、 设备连接鉴权 5](#_Toc11830289)

[四、 设备数据上报 6](#_Toc11830290)

[五、 设备命令处理（物模型类型产品） 9](#_Toc11830291)

[六、 在线调试设备（物模型类型产品） 11](#_Toc11830292)

[7.1云端设备上报数据查看 11](#_Toc11830293)

[7.2云端下发命令至设备 11](#_Toc11830294)

# 平台创建MQTT产品

1. 用户登录OneNET，如下图所示创建产品。



图1.1创建产品

1. 进入产品详情获取**产品ID**和**产品MasterKey。**



图1.2 获取产品信息

1. 新增功能定义（物模型产品类型）



图1.3 功能定义

1. 上传数据解析脚本（透传数据类型）

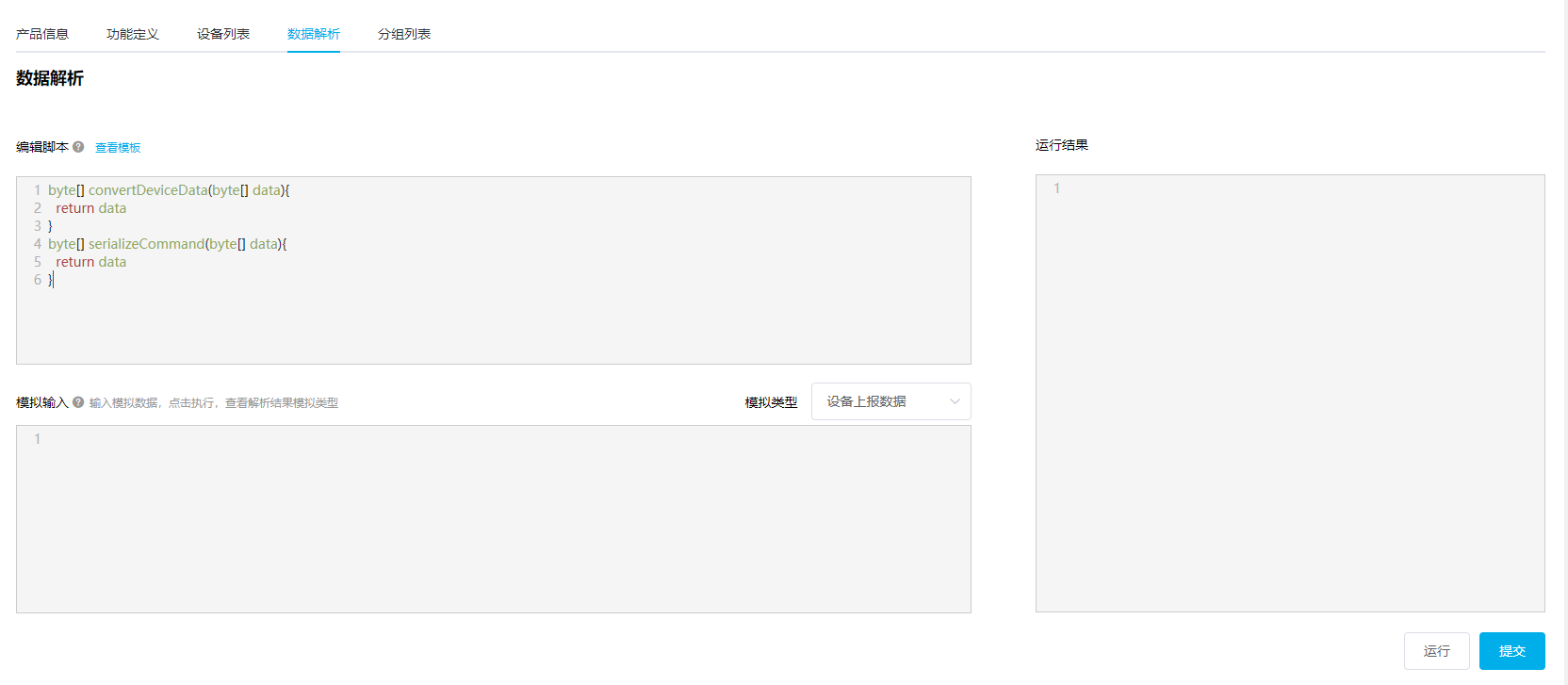


图1.4 上传脚本

注：目前只支持groovy脚本。

# 平台创建MQTT设备

1. 新增MQTT设备



图2.1 创建设备

2、进入设备详情获取**设备ID**



图2.1 获取设备ID

# 设备连接鉴权

* 设备向平台发起connect请求，connect中携带鉴权信息

表3.1 鉴权信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 信息名称 | 功能描述 | 对应属性值 | 是否必需 |
| Username | 用户名 | 产品ID | 是 |
| Password | 密码 | 产品MasterKey | 是 |
| ClientId | 客户端ID | 设备ID | 是 |
| QoS | 服务质量 | 默认为0 |  |
| CleanSession | 设置是否清除会话 | 默认为true |  |

- 若Username、Password、ClientId信息错误，平台将拒绝设备接入

- 若Username、Password、ClientId信息正确，设备可正常进行订阅或发布属于设备权限的Topic

# 设备数据上报

设备使用publish报文来上传数据点：

**平台需定义相应的功能定义，设备数据点才能正常上报，数据点名称即为属性名称。**

1. 设备上传数据Topic说明

合法设备端（即鉴权信息正确，已连接成功）仅能通过特定Topic上传数据，若格式错误，则无法上传数据至平台

Topic格式：**{producttype}/data/pid/{pid} /devkey/{deviceId}**

Topic参数说明：见下表：

表4.1 设备端上报数据Topic说明表

|  |  |
| --- | --- |
| 参数 | 参数说明 |
| producttype | 产品类型：  标准数据类型：standard  透传数据类型：customized |
| pid | 产品ID |
| deviceId | 设备ID |

Topic示例：standard/data/pid/176/devkey/135

1. 设备上传数据Payload说明

Payload包含真正的设备数据内容，支持格式如

数据类型2(type == 2)格式说明：

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte 1 | 数据点类型指示：type=2 //二进制数据 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| Byte 2 | *//指示后面json字符串长度*  固定两字节长度-高位字节，值为0x00 |  |  |  |  |  |  |  |  |
| Byte 3 | 固定两字节长度-低位字节，值为0x10 |  |  |  |  |  |  |  |  |
| Byte 4 | {  ”ds\_id”:”image”, *//创建数据流时定义的ID，（必填）*  “at”:”2014-10-25 12:23:23”, *//时间，（可选）*  ”desc”:字符串或json对象*//对该数据的描述（可选）*  } |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| … |  |  |  |  |  |  |  |  |
| Byte n |  |  |  |  |  |  |  |  |
| Byte n+1 | *//指示后面二进制数据长度*  固定四字节长度-第1字节(最高)，值为0x00 |  |  |  |  |  |  |  |  |
| Byte n+1 | 固定四字节长度-第2字节，值为0x00 |  |  |  |  |  |  |  |  |
| Byte n+2 | 固定四字节长度-第3字节，值为0x01 |  |  |  |  |  |  |  |  |
| Byte n+3 | 固定四字节长度-第4字节(最低)，值为0x00 |  |  |  |  |  |  |  |  |
| Byte n+4 | *//该域目前最大支持3M*  本例中的该域256字节数据 |  |  |  |  |  |  |  |  |
| … |  |  |  |  |  |  |  |  |
| Byte n+260 |  |  |  |  |  |  |  |  |

数据类型3(type == 3)格式说明：

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte 1 | 数据点类型指示：type=3 *// JSON格式2字符串* | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| Byte 2 | *//指示后面字符串长度*  固定两字节长度高位字节，值为0x00 |  |  |  |  |  |  |  |  |
| Byte 3 | 固定两字节长度低位字节，值为0x46 |  |  |  |  |  |  |  |  |
| Byte 4 | 通用格式：  {  “datastream\_id1”:”value1”,  “datastream\_id2”:”value2”,  …  }  示例：  {“temperature”:22.5,”humidity”:”95.2%”} |  |  |  |  |  |  |  |  |
| …  …  …  … |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| Byte n |  |  |  |  |  |  |  |  |

数据类型5(type == 5)格式说明

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte 1 | 数据点类型指示：type=5 *//自定义分隔符* | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| Byte 2 | *//指示后面字符串长度*  固定两字节长度高位字节，值为0x00 |  |  |  |  |  |  |  |  |
| Byte 3 | 固定两字节长度低位字节，值为0x41 |  |  |  |  |  |  |  |  |
| Byte 4 | 消息中最前面两字节为用户自定义的域中分隔符和域间分隔符，这两个分隔符不能相同。比如采用逗号作为域中分隔符，分号作为域间分隔符的格式如下：  ,;feild0;feild1;…;feildn  其中，每个field格式支持3种：  field格式1： 3个子字段，分别是数据流ID,时间戳，数据值。通用格式：  Datastream\_id,datetime,value  field格式2： 2个子字段，分别是数据流ID和数据值，省略时间戳。通用格式：  Datastream\_id,value  field格式3： 1个子字段，省略了数据ID和时间戳，只传输数据值，平台将用该域(feild)所在的位置号（从0开始）作为数据流ID。通用格式：  value  示例：  (1),;temperature,2015-03-22 22:31:12,22.5;102;pm2.5,89;10  (2)#@temperature#2015-03-22 22:31:12#22.5@102@pm2.5#89@10 |  |  |  |  |  |  |  |  |
| …  …  …  …  …  …  …  …  …  … |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| Byte n |  |  |  |  |  |  |  |  |

数据类型8(type == 8)格式说明

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte 1 | 数据点类型指示：type=8 *//自定义分隔符* | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| Byte 2 | *//指示后面字符串长度*  固定两字节长度高位字节，值为0x00。不含CRC16 |  |  |  |  |  |  |  |  |
| Byte 3 | 固定两字节长度低位字节，值为0x41。不含CRC16 |  |  |  |  |  |  |  |  |
| Byte 4 | 消息中最前面两字节为用户自定义的域中分隔符和域间分隔符，这两个分隔符不能相同。比如采用逗号作为域中分隔符，分号作为域间分隔符的格式如下：  ,;feild0;feild1;…;feildn  其中，每个field格式支持3种：  field格式1： 3个子字段，分别是数据流ID,时间戳，数据值。通用格式：  Datastream\_id,datetime,value  field格式2： 2个子字段，分别是数据流ID和数据值，省略时间戳。通用格式：  Datastream\_id,value  field格式3： 1个子字段，省略了数据ID和时间戳，只传输数据值，平台将用该域(feild)所在的位置号（从0开始）作为数据流ID。通用格式：  value  示例：  (1),;temperature,2015-03-22 22:31:12,22.5;102;pm2.5,89;10  (2)#@temperature#2015-03-22 22:31:12#22.5@102@pm2.5#89@10 |  |  |  |  |  |  |  |  |
| …  …  …  …  …  …  …  …  …  … |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| Byte n |  |  |  |  |  |  |  |  |
| Byte n+1 | CRC16，计算Byte 4到Byte n的CRC16值 |  |  |  |  |  |  |  |  |

# 设备命令处理（物模型类型产品）

云端采用MQTT publish方式下发平台指令至设备端：

1. 设备端命令Topic

合法设备端（即鉴权信息正确，已连接成功）仅能通过订阅特定Topic接收命令,并通过特定Topic 发布命令ACK。

命令订阅Topic格式：**{producttype}/cmd/pid/{pid}/devkey/{deviceId}**

命令回复Topic格式：**{producttype}/resp/pid/{pid}/devkey/{deviceId}**

Topic参数说明：见下表：

表5.1 设备端上报数据Topic说明表

|  |  |
| --- | --- |
| 参数 | 参数说明 |
| producttype | 产品类型：  标准数据类型：standard  透传数据类型：customized |
| pid | 产品ID |
| deviceId | 设备ID |

命令订阅Topic示例：standard/cmd/pid/176/devkey/135

命令回复Topic示例：standard/resp/pid/176/devkey/135

1. 设备接收命令Payload以及回复Payload说明

命令Payload以及回复Payload皆为标准json格式，回复Payload中uuid值需和接收的命令一致，进行唯一匹配。

2.1、读命令Payload说明

命令订阅Topic示例：standard/cmd/pid/176/devkey/135

命令格式：

|  |  |
| --- | --- |
| Key | Value说明 |
| cmd | 功能定义中属性名 |
| method | 读命令为“get” |
| uuid | 命令唯一ID，命令回复中需携带此ID |

命令示例：

**{**

**"cmd":"shuxing",**

**"method":"get",**

**"uuid":"127ac9b9-341a-45a2-a7e0-039e3c4def30"**

**}**

命令回复Topic示例：standard/resp/pid/176/devkey/135

回复格式：

|  |  |
| --- | --- |
| Key | Value说明 |
| uuid | 命令唯一ID，填写接收命中的uuid |
| 属性名 | 设备回复属性值 |

回复示例：

**{**

**"uuid":"127ac9b9-341a-45a2-a7e0-039e3c4def30",**

**"shuxing ":"devvalue"**

**}**

2.2 写命令payload说明

命令订阅Topic示例：standard/cmd/pid/176/devkey/135

命令格式：

|  |  |
| --- | --- |
| Key | Value说明 |
| cmd | 功能定义中属性名 |
| method | 读命令为“set” |
| uuid | 命令唯一ID，命令回复中需携带此ID |
| 属性名 | 平台设置属性值 |

命令示例：

**{**

**"cmd":"shuxing",**

**"method":"get",**

**"uuid":"5853ad02-60bc-4745-bbd7-c05e80f1cee4",**

**"shuxing ":"setvalue",**

**}**

命令回复Topic示例：standard/resp/pid/176/devkey/135

回复格式：

|  |  |
| --- | --- |
| Key | Value说明 |
| uuid | 命令唯一ID，填写接收命中的uuid |

回复示例：

**{**

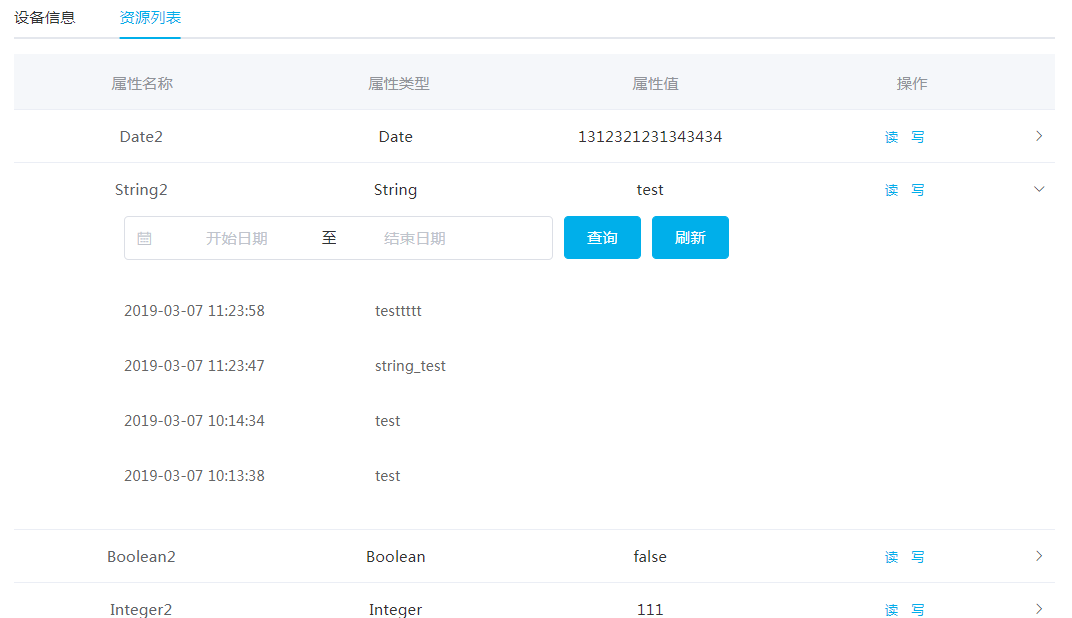
**"uuid":"5853ad02-60bc-4745-bbd7-c05e80f1cee4",**

**}**

# 在线调试设备（物模型类型产品）

## 6.1云端设备上报数据查看

在设备资源列表中可以看到设备主动上报的数据以及下发命令主动读取的数据。



## 6.2云端下发命令至设备

可以通过设备资源列表中的读写命令下发命令至设备。

