
Parser 文档

2018-02-07



百度云
cloud.baidu.com

目录

1	产品描述	1
1.1	介绍	1
1.2	优势	2
1.3	核心概念	2
1.4	功能	3
1.5	应用场景	3
2	操作指导	4
2.1	操作准备	4
2.1.1	账号注册及实名认证	4
2.1.2	准备企业网关设备	4
2.1.3	操作流程介绍	4
2.2	安装IoT Edge SDK	5
2.3	管理网关设备	6
2.3.1	新建网关	6
2.3.2	查看密钥	7
2.3.3	重置密钥	8
2.3.4	禁用网关	8
2.3.5	管理子设备	9
	新建网关子设备	9
2.4	通过网关反控设备	11
2.5	解析项目	13

2.5.1	新建解析项目	13
2.5.2	设置解析参数	14
	设置Modbus通信地址表	14
	设置轮询请求	15
	停止解析	16
2.5.3	暂停解析项目	16
2.5.4	删除解析项目	17
2.6	附录	17
2.6.1	数学计算公式	17
3	产品定价	21
4	API参考	22
4.1	介绍	22
4.1.1	简介	22
4.1.2	调用方式	22
	概述	22
	通用约定	22
	公共请求头	23
	公共响应头	23
	响应状态码	23
	通用错误返回格式	23
	公共错误码	24
	签名认证	24
	签名生成算法	25
4.1.3	多区域选择	25
4.2	API接口描述	25
4.2.1	headers	25
4.2.2	排序 (order)	25

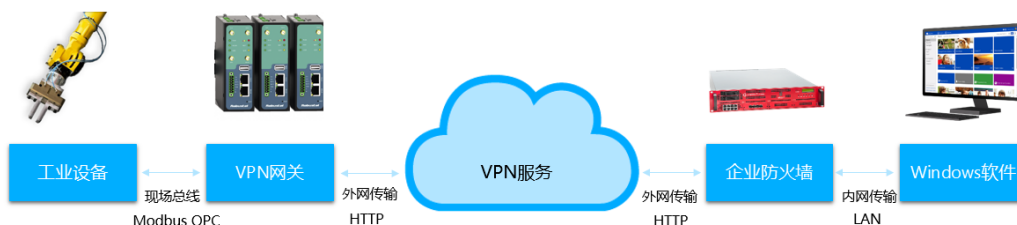
4.2.3	分页 (pagination)	26
4.2.4	搜索 (search)	26
4.3	Gateway	26
4.3.1	分页查询	27
4.3.2	获取	28
4.3.3	创建	29
4.3.4	更新	30
4.3.5	重置密钥	31
4.3.6	删除	32
4.4	Device	32
4.4.1	筛选	33
4.4.2	获取	34
4.4.3	创建	35
4.4.4	更新	37
4.4.5	删除	38
4.5	ParserObject	38
	筛选	39
4.5.1	获取	40
4.5.2	创建	41
4.5.3	更新	42
4.5.4	删除	43
4.6	PullRule	44
4.6.1	筛选	44
4.6.2	获取	46
4.6.3	创建	47
4.6.4	更新	48
4.6.5	删除	48
4.7	DataDescription	49

4.7.1	筛选	49
4.7.2	获取	51
4.7.3	创建	52
4.7.4	更新	53
4.7.5	删除	54
4.8	Action	54
4.8.1	下发配置	54
4.9	Custom Property	55
4.9.1	筛选	55
4.9.2	添加	55
4.9.3	删除	56
5	常见问题	57
5.1	物解析常见问题	57
5.1.1	物解析和物接入的关系是什么？	57
5.1.2	对网关配置的轮询策略是什么时候写入网关的？	57
5.1.3	点击“全部生效”时，写入网关的轮询策略是什么？	57
5.1.4	为什么不能新建轮询设置？	57
5.1.5	暂停或启动某个解析项目，云端和网关端是如何工作的？	57
5.1.6	禁用或恢复某个网关，云端和网关端是如何工作的？	58

第1章 产品描述

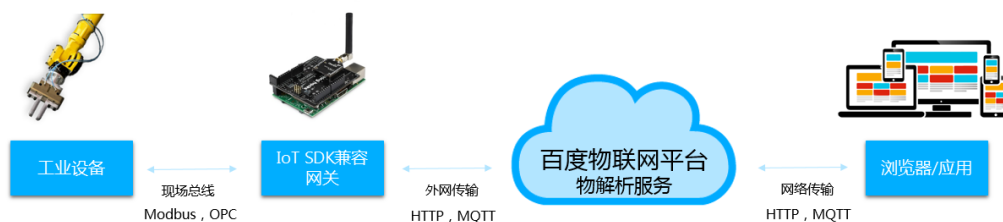
1.1 介绍

为了更便捷的管理和维护生产设备，企业需要大量采集、存储和分析生产数据，用数据中得到的洞察来帮助工厂优化生产和设备。传统情况下，为了把生产数据从工厂传送至企业内部，企业需要部署VPN网关、购买VPN网络服务并搭配Windows软件，才能连接到远端设备，获取并解析处理相关数据信息，如下图所示：



传统方式下，企业需要购买专业的设备和服务，并对整个网络进行维护，IT成本高，且无法应对频繁的网络变化。如果工厂不止一个，不同工厂之间的数据在传统模式下，会形成一个个信息孤岛；由于没有统一的数据存储中心，可能得到片面的分析结果，没法做到真正的协同生产。

百度云物联网平台是为转型升级中的工业企业和工业解决方案提供商提供的领先可靠的物联网平台。通过百度云物联网平台提供的“物接入”和“物解析”服务，用户可以快速将工业设备安全接入云端，同时将设备的报文转换为可提供分析洞察的数据，如下图所示：



物解析就是在云端为用户提供工业协议解析服务（比如Modbus和OPC UA）服务。当云端收到网关返回的原始数据后，结合用户提供的设备通讯地址表，“物解析”将数据解析成直接

可用于存储和分析的数据。

1.2 优势

解析更高效，配置更灵活

把网关和Windows软件端进行的协议报文（比如Modbus和OPC UA）解析工作，合二为一，移到了云端。

降低开发周期，提高部署能力

以前需要编写代码完成解析，现在只需要填写配置。无须硬件端软件修改，提高部署效率。

成本更低，弹性扩展

支持成本更低的网关，存储和计算均为分布式架构，支持弹性扩展。

1.3 核心概念

Modbus通讯协议：Modbus协议是应用于电子控制器上的一种通用语言。通过此协议，控制器相互之间、控制器经由网络（例如以太网）和其它设备之间可以通信。通过Modbus，不同厂商生产的控制设备可以连成工业网络，进行集中监控。

Modbus协议模式：物接入按照Modbus协议支持3种模式，TCP, RTU, ASCII。目前物解析支持TCP和RTU两种模式。

- TCP模式，Modbus主站和从站之间通过TCP协议通信。
- RTU模式：Modbus主站和从站之间通过串口连接，如RS232, RS485。数据包里面数据是二进制格式，没有经过任何编码，传输效率最高。
- ASCII模式：Modbus主站和从站之间通过串口连接，如RS232, RS485。数据包里面数据是将二进制数据进行ASCII编码，每个byte编码成2个字符，如byte 0x4f会编码成'4f' 2个字符进行传输。传输效率相对RTU偏低，好处是可以在传输线路上监控可读的数据。

MQTT：MQTT是基于二进制消息的发布/订阅（Publish/Subscribe）模式的协议，最早由IBM提出的，如今已经成为OASIS规范，更符合M2M大规模沟通。

网关设备：网关设备是连接企业内部设备（比如PC、PLC、SCADA、DCS）与企业外网云服务的桥梁，它接受设备轮询配置管理服务下发的配置、执行设备轮询配置的读取策略、并在收到设备返回的数据后，将数据上传回云端。

网关子设备：指某网关设备所在网络内与之连接的工业设备，工业设备采集的数据将经该网关上传至云端。

1.4 功能

网关设备管理

用户通过可视化Web UI来管理企业网关，以及企业网关所连接的子设备。在云端添加新的网关（对应厂区的工业网关），会自动生成对应新的连接信息和连接密钥。厂区的工业网关可以通过配置密钥来完成与云端连接。

IoT Edge SDK

在企业网关在集成IoT Edge SDK后，只需要配置一个云端生成的密钥便可以完成与云端连接，实现与云端通讯配置。IoT Edge SDK包含MQTT、Modbus等相关组件、以及与配置服务通信的基本实例代码。

设备轮询配置管理

设备轮询配置管理会把用户配置的信息通过MQTT协议通道下发到企业网关。用户可通过云端控制台的可视化Web UI操作或者Restful API调用方式选择要向哪些设备发送轮询配置。

协议数据解析

数据解析负责把网关上发到云端的设备原始数据（一般为未解析的二进制数据包）结合设备的通讯地址表（也称为点表）数据进行数据解析，然后进行数据存储，以进行后续的分析工作。

1.5 应用场景

工业设备（比如机器手臂）通过PLC，SCADA，DCS等自动化控制系统采集的数据，可以经由工业通讯协议（比如Modbus，OPC UA，ProfiNet等），将原始数据通过工业网关在未经协议解析的情况下发送转发到云端，然后在云端服务将原始数据进行工业协议解析。同时对于设备的采集频率和采集配置，也由云端统一设置。这样可以极大的降低开发时间和成本，加快了工厂设备数据采集的部署速度成本，提高了协议解析采集效率和灵活性，同时该技术使用的分布式云端方案，也满足工业客户设备同时高并发接入时，需要的稳定性和弹性扩展。随着数据量的增长，云端支持的海量存储使企业无须再担心本地存储不够，以及对海量数据存储的需要。同时百度云提供一系列大数据工具，为这些海量数据创造价值提供有力的支持。

第2章 操作指导

2.1 操作准备

2.1.1 账号注册及实名认证

注意：

推荐用户使用Firefox或Chrome浏览器执行控制台操作。

在操作专属服务器前，应先完成以下任务：

- 完成百度云[账号注册](#)。
- 完成百度云[实名认证](#)。

2.1.2 准备企业网关设备

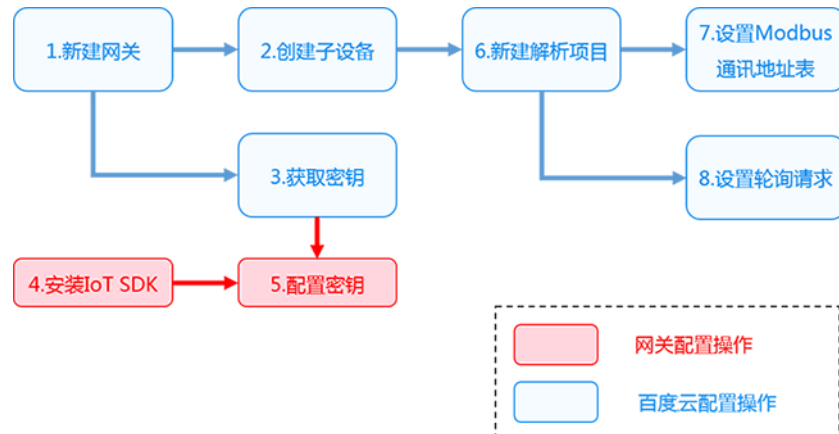
网关设备是连接企业内部设备（比如PC、PLC、SCADA、DCS）与企业外网云服务的桥梁，它接受设备轮询配置管理服务下发的配置、执行设备轮询配置的读取策略、并在收到设备返回的数据后，将数据上传回云端。

为了实现网关设备与百度的对接，需要在网关设备上安装IoT Edge SDK。在企业网关在集成IoT Edge SDK后，只需要配置一个云端生成的密钥便可以完成与云端连接，实现与云端通讯配置。IoT Edge SDK包含MQTT、Modbus等相关组件、以及与配置服务通信的基本实例代码。

在部署物解析服务之前，请先确认网关设备提供商确认企业的网关设备是否可以安装SDK，具体操作方法请查看[安装IoT Edge SDK](#)。

2.1.3 操作流程介绍

物解析的操作与配置流程如下图所示：



1. **新建网关**:在云端创建云网关，作为厂区网关在云端的一对一的映射。
2. **创建子设备**：网关子设备是指与网关设备连接的工业设备。请在网关设备下新建网关子设备，作为厂区工业设备的一对一映射。
3. **获取密钥**：新建网关后，系统将自动为网关分配密钥。
4. **安装IoT SDK**：为了实现网关设备与百度的对接，需要在网关设备上安装IoT Edge SDK。
5. **配置密钥**：为了使厂区网关能正确上传数据到云网关，需要将系统生成的密钥配置到对应的厂区网关中。操作方法请查看**安装IoT SDK**中的具体操作步骤。
6. **新建解析项目**：负责对网关上发到云端的设备原始数据进行解析。
7. **设置Modbus通讯地址表**：解析服务将根据通讯地址表的内容，将Modbus消息解析成可用的数据和内容。
8. **设置轮询请求**：通过设置轮询请求指定网关设备向网关子设备请求数据的地址和周期间隔。

2.2 安装IoT Edge SDK

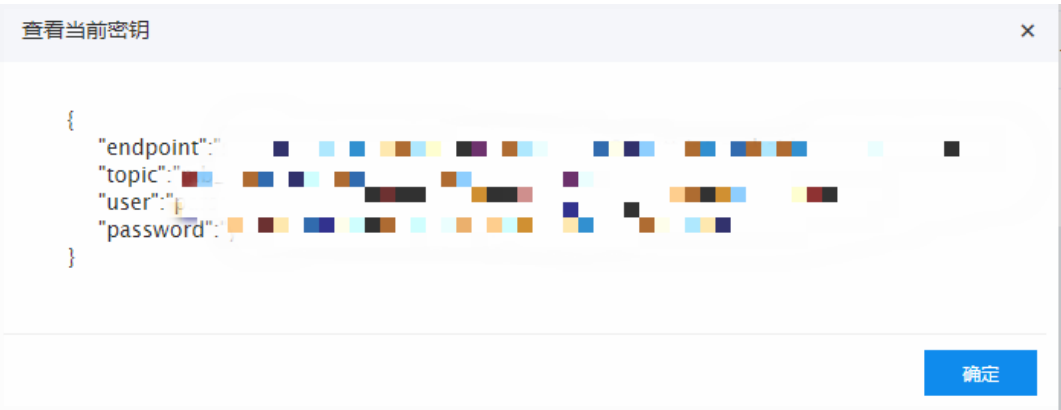
目前，我们在**IoT Edge SDK**提供了多个安装脚本，它们分别是：

- Modbus/nossl/install-ubuntu.sh: 该脚本在ubuntu下编译运行于linux下的不支持SSL加密通信的网关；
- Modbus/withssl/install-ubuntu.sh: 该脚本在ubuntu下编译运行于linux下的支持SSL加密通信的网关；
- Modbus/corss-compile: 该目录提供了一批在ubuntu下编译，运行于其他平台（linux-arm, win32, win64）的支持SSL加密的网关；

注意：网关设备可以通过SSL加密方式或非加密方式与云端连接，请根据网关设备的实际情况选择。

1. 在bdModbusGateway所在目录下创建文件名为gwconfig.txt的配置文件，bdModbusGateway的运行依赖于配置文件gwconfig.txt，在云端新建网关后，系统会自动为网关

分配密钥，请将该json格式的连接信息（含密钥）写入gwconfig.txt，如下图所示。关于密钥的获取方法，请参看[查看密钥](#)。



- 2. 运行bdModbusGateway。运行后，网关会一直订阅gwconfig.txt中配置的MQTT主题，从而接受从云端上配置的Modbus从站轮询策略。轮询策略指定了bdModbus-Gateway获取信息的频率、子设备等信息。

2.3 管理网关设备

2.3.1 新建网关

在云端创建云网关，作为厂区网关在云端的一对一的映射。具体操作方法如下：

- 1. 登录[百度云官网](#)，点击右上角的“管理控制台”，快速进入控制台界面。
- 2. 选择“产品服务>物解析 IoT Parser>网关设备管理”，进入“网关设备管理”页面。
- 3. 点击“新建网关”，在弹出对话框中输入“网关ID”及“网关描述”。“网关ID”仅支持英文、数字，且必须唯一。为了便于网关设备管理，“网关ID”应尽量有实际意义。如果用户网络中有多台网关设备，应提前做好ID规划。





4. 返回“网关设备管理”页面，可以从网关列表中查看相关配置信息。



相关操作

为了使厂区网关能正确上传数据到云网关，需要将系统生成的密钥配置到对应的厂区网关中，获取密钥信息的方法请参看[查看密钥](#)。

新建网关后，需要继续添加接入该网关的工业设备，具体操作请参看[新建网关子设备](#)。

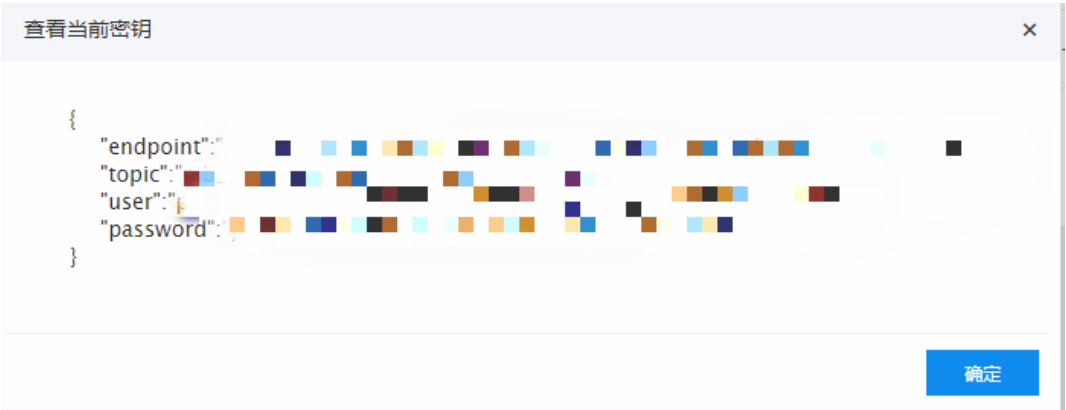
2.3.2 查看密钥

为了使厂区网关能正确上传数据到云网关，需要将系统生成的密钥配置到对应的厂区网关中，网关密钥的查询方法如下：

- 1. 选择“产品服务>物解析 IoT Parser>网关设备管理”，进入“网关设备管理”页面。
- 2. 找到网关设备，点击“更多操作>查看密钥”。



- 3. 将json格式的连接信息保存至网关设备的配置文件gwconfig.txt中，关于配置文件gwconfig.txt的详细介绍，请参看[安装IoT Edge SDK](#)。



2.3.3 重置密钥

注意：

重置密钥后，旧密钥将自动失效，请及时修改网关设备的配置文件，否则网关设备无法将数据信息上传到云端。

- 1. 选择“产品服务>物解析 IoT Parser>网关设备管理”，进入“网关设备管理”页面。
- 2. 找到网关设备，点击“更多操作>重置密钥”。

<div>+ 新增网关 删除</div>						
网关ID	状态	网关描述	子设备	创建时间	操作	
TEST02	在线	演示示例02	0	2017-08-11 11:27:19	管理子设备 配置下发 更多操作	禁用
TEST01	在线	演示示例01	0	2017-08-02 15:31:09	管理子设备 配置下发	重置密钥
111	在线	111	0	2016-10-17 16:11:04	管理子设备 配置下发	重置密钥

- 3. 点击“查看密钥”，将json格式的连接信息保存至网关设备的配置文件gwconfig.txt中，关于配置文件gwconfig.txt的详细介绍，请参看[安装IoT Edge SDK](#)。
- 4. 点击“确定”，使新密钥设置生效。

网关描述	子设备	创建时间	操作
演示示例02	1	2017-08-11 11:27:19	管理子设备 配置下发 更多操作
演示示例01	0	2017-08-02 15:31:09	管理子设备 配置下发 更多操作
		2016-10-17 16:11:04	管理子设备 配置下发 更多操作

提示

重置密钥后，网关连接MQTT的密钥将更改，可能需要重新烧录程序，请谨慎操作。是否重置密钥？

确定 取消

2.3.4 禁用网关

注意：
禁用网关后，网关将停止一切数据上传工作。

1. 选择“产品服务>物解析 IoT Parser>网关设备管理”，进入“网关设备管理”页面。
2. 找到网关设备，点击“更多操作>禁用”。

+ 新建网关 删除						
网关ID	状态	网关描述	子设备	创建时间	操作	
TEST02	● 在线	演示示例02	0	2017-08-11 11:27:19	管理子设备 配置下发	更多操作 ^
TEST01	● 在线	演示示例01	0	2017-08-02 15:31:09	管理子设备 配置下发	禁用 重置密钥
111	● 在线	111	0	2016-10-17 16:11:04	管理子设备 配置下发	查看详情

3. 点击“确定”，使网关禁用操作生效。

+ 新建网关 删除						
网关ID	状态	网关描述	子设备	创建时间	操作	
TEST02	● 在线	演示示例02	1	2017-08-11 11:27:19	管理子设备 配置下发	更多操作 ^
TEST01	● 在线	演示示例01	0	2017-08-02 15:31:09	管理子设备 配置下发	更多操作 ^
111	● 在线	111	0	2016-10-17 16:11:04	管理子设备 配置下发	更多操作 ^

禁用网关

ⓘ 禁用网关后，需要点击『配置下发』，将禁用指令下发给网关才能生效。是否确认禁用该网关？

确定 取消

网关被禁用后，网关状态将显示为“禁用”。此时可点击“更多操作>恢复>配置下发”重新启动网关。

禁用或恢复某个网关，云端和网关端是如何工作的？

2.3.5 管理子设备

新建网关子设备 网关子设备是指与网关设备连接的工业设备。请在网关设备下新建网关子设备，作为厂区工业设备的一对一映射。

在新建网关子设备之前应先完成**新建网关**操作。

1. 选择“产品服务>物解析 IoT Parser>网关设备管理”，进入“网关设备管理”页面。
2. 找到网关设备，点击“管理子设备”，进入子设备管理页面。

+ 新建网关 删除						
网关ID	状态	网关描述	子设备	创建时间	操作	
TEST02	● 在线	演示示例02	0	2017-08-11 11:27:19	管理子设备 配置下发	更多操作 ^
TEST01	● 在线	演示示例01	0	2017-08-02 15:31:09	管理子设备 配置下发	更多操作 ^

3. 点击“新建网关子设备”，在弹出对话框中输入配置信息。



- * 模式，主站和从站之间通过协议通信。□ TCPModbusTCP
- * 模式：主站和从站之间通过串口连接，如RTUModbusRS232, 。RS485选择模式时需要配置以下信息：

TCP

- * 子设备：用来标识不同的网关子设备，仅支持英文、数字，同一个网关设备下的子设备必须唯一。IDID
- * 从站（）：从站编号，取值slaveModbus，在同一台网关设备下，每一个网关子设备需要拥有唯一的从站编号。[1–247]
- * 描述：对子设备进行补充描述。
- * 地址：子设备的地址信息。IPIP
- * 端口号：子设备的端口。Modbus选择模式时需要配置以下信息：

RTU

- * 子设备：用来标识不同的网关子设备，仅支持英文、数字，同一个网关设备下的子设备必须唯一。IDID
- * 从站（）：从站编号，取值slaveModbus，在同一台网关设备下，每一个网关子设备需要拥有唯一的从站编号。[1–247]
- * 描述：对子设备进行补充描述。
- * 串口号：即通信是经过哪个串口号进行。
- * 波特率：网关子设备的波特率。
- * 数据位：取值8或两种bits7bits
- * 校验：取值无校验(None)，奇校验(Odd)，或者偶校验(Even)。
- * 停止位：取值1，或者。bits2bits

4. 完成新建子设备操作后返回子设备管理页面，查看网关子设备信息。

TEST02 ● 在线

+ 新建网关子设备

删除

<input type="checkbox"/> 子设备ID	状态	从站(slave)	子设备描述	接入方式	操作
<input type="checkbox"/> TEST01	● 在线	100	演示子设备02	TCP	<div>修改禁用查看详情</div>

5. 如果需要更新网关子设备配置信息，可以点击“修改”进入编辑模式。

注意：

禁用网关子设备后，网关设备将无法获取该子设备的数据信息。

1. 选择“产品服务>物解析 IoT Parser>网关设备管理”，进入“网关设备管理”页面。
2. 找到网关设备，点击“管理子设备”，进入子设备管理页面。
3. 找到网关子设备，点击“禁用”，禁用该网关子设备。对于已经被禁用的网关设备，用户可点击“恢复”重新启用该设备。

TEST02 ● 在线

+ 新建网关子设备

删除

<input type="checkbox"/> 子设备ID	状态	从站(slave)	子设备描述	接入方式	操作
<input type="checkbox"/> TEST01	● 在线	100	演示子设备02	TCP	修改 禁用 查看详情

注意：

网关子设备被删除后无法恢复，请谨慎操作。

1. 选择“产品服务>物解析 IoT Parser>网关设备管理”，进入“网关设备管理”页面。
2. 找到网关设备，点击“管理子设备”，进入子设备管理页面。
3. 勾选需要删除的网关子设备（可复选），点击“删除”完成网关子设备删除操作。

TEST02 ● 在线

+ 新建网关子设备

删除

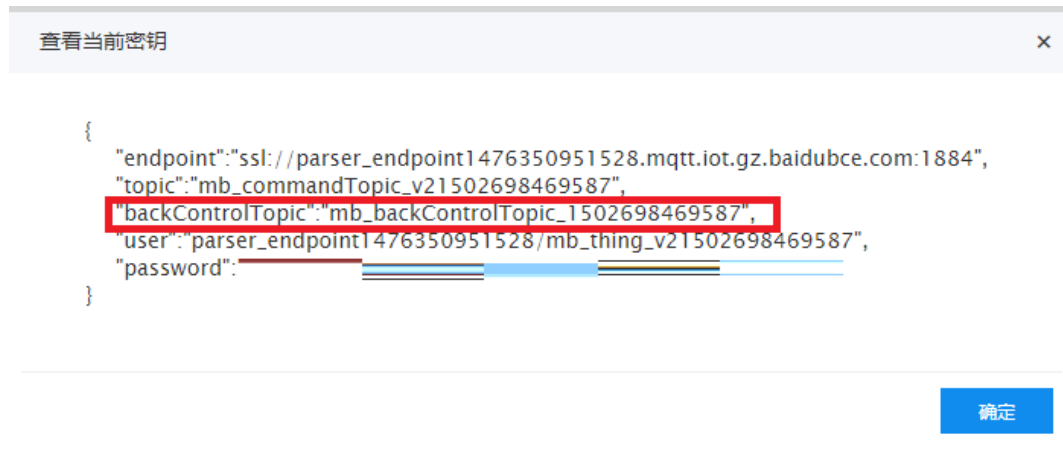
<input checked="" type="checkbox"/> 子设备ID	状态	从站(slave)	子设备描述	接入方式	操作
<input checked="" type="checkbox"/> TEST01	● 在线	100	演示子设备02	TCP	修改 禁用 查看详情

2.4 通过网关反控设备

物解析的网关包括backControlTopic字段，用来反控网关行为，即物解析网关不仅可以从Modbus采集数据上传到云端，还可以从云端接受反控指令，写入Modbus从站。

注意：网关需要下载新版(2017年7月24日之后)，[下载地址](#)。

首先，[新建网关](#)，点击[查看密钥](#)，此时可以看到backControlTopic的字段，配置到最新的网关，网关订阅这个topic后，用户就可以向这个topic发送反控指令。网关订阅到反馈指令，就会去执行，即向指定的Modbus从站写指定的数据。



当用户需要反控的时候，就向backControlTopic发送一条反控指令，反控指令为JSON格式，示例如下：

```
{
  "request1" : {
    "slaveid" : 1,
    "address" : 1,
    "data" : " 0101010100000000"
  },
  "request2" : {
    "slaveid" : 2,
    "address" : 40001,
    "data" : " 00ff1234"
  }
}
```

requestx: x为数字编号,1,2,3...n, 表示一个消息包含的多个指令。 slaveid: 为需要反控的modbus从站编号。 address: 为需要写的寄存器的起始地址。 data: 为要写入modbus从站的数据。

从address开始，依次往后写入。例如示例中的request1，会向地址1-8等8个离散值(coins)写数据，写入的值分别为1,1,1,0,0,0,0；示例中的request2，会写2个寄存器，40001和40002，写入的值分别为00ff, 1234。

注意：反控指令不需要指定功能码(function code)，这里的address是绝对地址，也就是对应解析项目里面解释设置（点表）的地址，而不是轮询设置里面的起始地址。需要反控的Modbus从站，必须是正在被采集的从站，也就是说，轮询设置里面有针对该从站的轮询。例如：如果网关采集的slaveid为1.2,即可以控制slaveid为1.2的从站，反

控slaveid=3的从站，这是不被允许的。

如果你不希望网关能反控你的Modbus从站，你可以在网关配置文件(gwconfig.txt)里面，将backControlTopic删除。关于配置文件gwconfig.txt的详细介绍，请参看[安装IoT Edge SDK](#)。

2.5 解析项目

数据解析负责把网关上发到云端的设备原始数据（一般为未解析的二进制数据包）结合设备的通讯地址表（也称为点表）数据进行数据解析，然后进行数据存储，以进行后续的分析工作。

2.5.1 新建解析项目

在新建解析项目前，应先完成以下操作：

- [新建网关](#)和[新建网关子设备](#)作为解析数据的来源。
- [创建BOS Bucket](#)存储解析后的数据。

新建解析项目的具体操作步骤如下：

1. 选择“产品服务>物解析 IoT Parser>解析项目”，进入“解析项目”操作界面。
2. 点击“新建解析项目”，在弹出对话框中输入相关配置信息并点击“确定”完成解析项目创建。

解析项目

解析项目配置完成后，需要配置下发才能生效，请到网关列表中点击相应网关的【配置下发】，完成配置的生效。

<div><div>+ 新建解析项目</div><div>删除</div></div>							
<input type="checkbox"/> 项目名称	状态	协议	数据来源	存储位置	目的地主题	创建时间	操作
<input type="checkbox"/> test	● 运行中	MODBUS	111	bos://1234-a/		2016-11-23 13:51:41	暂停 编辑 解析设置 数据验证

* 项目名称：指定解析项目名称，用来标识具体解析项目。

* 工业协议：目前仅支持协议。Modbus

* 数据来源：指定解析数据来源于哪个网关设备。用户可从下拉菜单中选择网关设备的设备。ID

* 数据存储：指定解析后的数据的存储位置（名称）。Bucket

* 目的地主题：指定解析后的数据转发到哪个主题。MQTT

3. 返回操作界面，可以查看到新建的解析项目。



4. 返回“网关设备管理”页面，点击“配置下发”使相关配置生效。

2.5.2 设置解析参数

在设置解析参数之前，应先完成新建解析项目操作。

设置Modbus通信地址表 解析服务将根据通讯地址表的内容，将Modbus消息解析成可用的数据和内容。

- 1. 选择“产品服务>物解析 IoT Parser>解析项目”，进入“解析项目”操作界面。
- 2. 找到指定的解析项目，点击“解析设置”，进入该解析项目的设置界面。



3. 点击“Modbus通讯地址表”页签，根据实际情况选择数据区，包括：保持寄存器和线圈状态。

告警	告警状态	详情
01 线圈状态 (Coil Status) (0x)	单个比特	00001 ~ 09999
02 输入状态 (Input Status) (1x)	单个比特	10001~19999
03 保持寄存器 (Holding Register) (4x)	16-比特字	40001 ~ 49999
04 输入寄存器 (Input Register) (3x)	16-比特字	30001~39999

test ● 运行中

Modbus通讯地址表 轮询请求设置

02 保持寄存器(Holding Register) (4x) 01 线圈状态(Coil Status) (0x) 02 输入状态 (Input Status) (1x) 04 输入寄存器 (Input Register) (3x)

+ 新建记录 批量导入 批量导出 解析服务根据通讯地址表的内容，将Modbus消息解析成可直接使用的数据和内容。

地址(十进制) ↓	显示名称	BIT位	数据类型	数据长度	单位	量程上限(RH)	量程下限(RL)	计算公式	其他	操作
44888	test	11	INT	16		10000	10	[?]	[?]	不解析 编辑 删除

4. 点击“新建记录”，设置相关参数，关于参数的具体介绍如下：

- 地址：Modbus设备内存中的地址，对于保持寄存器，地址范围是40001~49999之间；对于线圈状态，地址范围是00001~09999之间。对于功能码FC=3，起始地址=0，对应的寄存器地址为40001。
- 比特位：主要用于保持寄存器，由于保持寄存器存储的是16-比特字，如果将BOOL变量（开关变量）储在保持寄存器数据段，则需要说明BOOL变量具体放在16-比特字中的第几个bit，0表示该寄存器最右边的bit(least significant bit), 15表示最左边的bit(most significant bit)。
- 显示名称：说明数据的含义。解析服务在解析后会把名称附在数据上，方便理解。英文字符、数字和下划线。长度[3-40]，并且以英文字符开头。
- 量程上限：用户自定义的数据，代表该项目的最大值，通常用于计算公式里面带入的常量。可选参数，如果计算公式里面用到了“RH”变量，则需要定义量程上限。
- 量程下限：用户自定义的数据，代表该项目的最小值，通常用于计算公式里面带入的常量。可选参数，如果计算公式里面用到了“RL”变量，则需要定义量程下限。
- 数据类型：数据解析的目标类型。如INT，BOOL，REAL等。如果是计数寄存器，那么就是INT；如果是温度，则可能是REAL。
- 数据长度：从地址开始，数据占多少个bit。如果该数据占2个字节，那么这里选16。
- 单位：数据的单位。例如：压力，可选”MPa”，如果是计数，可不选。单位会附在解析后的数据上，方便理解。
- 计算公式：在物解析服务把二进制流解析成目标类型(INT, REAL, BOOL)之后，用户可以通过自定义一个公式来计算最终希望的数据。公式中X代表原始二进制数据解析后值，公式中可以引用RH, RL。例如：“(x-6400)/25600*(RH-RL)+RL”。计算公式中可以使用的运算符及函数请参看[附录-数学计算公式](#)

5. 返回设置界面，查看新增解析记录。

test ● 运行中

Modbus通讯地址表 轮询请求设置

02 保持寄存器(Holding Register) (4x) 01 线圈状态(Coil Status) (0x) 02 输入状态 (Input Status) (1x) 04 输入寄存器 (Input Register) (3x)

+ 新建记录 批量导入 批量导出 解析服务根据通讯地址表的内容，将Modbus消息解析成可直接使用的数据和内容。

地址(十进制) ↓	显示名称	BIT位	数据类型	数据长度	单位	量程上限(RH)	量程下限(RL)	计算公式	其他	操作
44888	test	11	INT	16		10000	10	[?]	[?]	不解析 编辑 删除

设置轮询请求 用户可通过设置轮询请求指定网关设备向网关子设备请求数据的地址和周期间隔，具体配置方法如下：

1. 选择“产品服务>物解析 IoT Parser>解析项目”，进入“解析项目”操作界面。

2. 找到指定的解析项目，点击“解析设置”，进入该解析项目的设置界面。

解析项目

解析项目配置完成后，需要配置下发才能生效，请到网关列表中点击相应网关的『配置下发』，完成配置的生效。

+ 新建解析项目

删除

项目名称	状态	协议	数据来源	存储位置	目的地主题	创建时间	操作
<input type="checkbox"/> test	● 运行中	MODBUS	111	bos://1234-a/		2016-11-23 13:51:41	暂停 编辑 解析设置 数据验证

3. 点击“轮询请求设置>新建轮询设置”页签，设置相关参数，具体内容解释如下。

解析项目测试01 ● 运行中

Modbus通讯地址表

轮询请求设置

+ 新建轮询设置

从站(slave)	子设备ID	子设备描述	操作码	请求间隔(秒)	操作
-----------	-------	-------	-----	---------	----



你还没创建任何实例
点击左上角按钮立即创建

- * 操作码：对子设备内存中的指定部分进行读取，包括：“保持寄存器”和“线圈状态”。03—01—
- * 开始地址：设备内存中的地址，取值范围~。如操作码为Modbus099993起始地址为，则对应的地址为。040001
- * 读取数据长度：从地址开始，读取的数据长度。如果操作码为“线圈状态”，此处单位为01—；如果操作码为“bit03保持寄存器”，此处单位为—16—字。bit
- * 请求间隔：每隔多长时间读取一次数据，单位秒。

4. 完成配置后返回“网关设备管理”页面，点击“配置下发”使相关配置生效。

停止解析 停止解析后，云端将不再根据该记录解析网关设备上传的数据。

1. 选择“产品服务>物解析 IoT Parser>解析项目”，进入“解析项目”操作界面。
2. 找到指定的解析项目，点击“解析设置”，进入该解析项目的设置界面。
3. 找到指定的解析记录，点击“不解析”，停止针对该条记录的解析。

test ● 运行中

Modbus通讯地址表

轮询请求设置

03 保持寄存器(Holding Register) (4x)

01 线圈状态(Coil Status) (0x)

02 输入状态 (Input Status) (1x)

04 输入寄存器 (Input Register) (3x)

+ 新建记录

批量导入

批量导出

解析服务根据通讯地址表的内容，将Modbus消息解析成可直接使用的数据和内容。

地址(十进制)	显示名称	BIT位	数据类型	数据长度	单位	量程上限(RH)	量程下限(RL)	计算公式	其他	操作
44888	test	11	INT	16		10000	10	[?]	[?]	不解析 编辑 删除

2.5.3 暂停解析项目

注意：

暂停解析项目后，云端将停止解析来自指定网关设备的所有数据。

1. 选择“产品服务>物解析 IoT Parser>解析项目”，进入“解析项目”操作界面。
2. 找到指定的解析项目，点击“暂停”，停止解析操作。

解析项目

解析项目配置完成后，需要配置下发才能生效，请到网关列表中点击相应网关的【配置下发】，完成配置的生效。

+ 新建解析项目		删除					
<input type="checkbox"/> 项目名称	状态	协议	数据来源	存储位置	目的地主题	创建时间	操作
<input type="checkbox"/> test	● 运行中	MODBUS	111	bos://1234-a/		2016-11-23 13:51:41	暂停 编辑 解析设置 数据验证

对于已经被暂停的解析项目，用户可点击“启动”重新启用该项目。

暂停或启动某个解析项目，云端和网关端是如何工作的？

2.5.4 删除解析项目

注意：

解析项目被删除后无法恢复，请谨慎操作。

1. 选择“产品服务>物解析 IoT Parser>解析项目”，进入操作界面。
2. 勾选需要删除的解析项目（可复选），点击“删除”，执行删除操作。

解析项目

解析项目配置完成后，需要配置下发才能生效，请到网关列表中点击相应网关的【配置下发】，完成配置的生效。

+ 新建解析项目		删除					
<input checked="" type="checkbox"/> 项目名称	状态	协议	数据来源	存储位置	目的地主题	创建时间	操作
<input checked="" type="checkbox"/> test	● 运行中	MODBUS	111	bos://1234-a/	alarmTopic	2016-11-23 13:51:41	暂停 编辑 解析设置 数据验证

3. 点击“配置下发”使相关配置生效。

2.6 附录

2.6.1 数学计算公式

数学运算符

运算符	描述
+	加法运算符
-	减法运算符
*	乘法运算符
/	除法运算符
%	取模运算符
^	指数运算符

位运算符

运算符	描述	示例
&	与	5&4=4
	或	2 1=3
^	异或	3^1=2
<<	左移	5<<1=10
>>	右移	5>>1=2

布尔运算符

运算符	描述
=	等于
==	等于
!=	不等于
<>	不等于
<	小于
<=	小于等于
>	大于
>=	大于等于
&&	布尔与运算
	布尔或运算

函数

运算符	描述
NOT(expression)	取反运算, NOT(0) = 1, NOT(非0) = 0

运算符	描述
IF(condition,value_if_true,value_if_false)	如果condition为真，返回value_if_true，否则返回value_if_false
RANDOM()	返回一个0 ~ 1之间的随机数
MIN(e1,e2, ...)	传回e1、e2、……较小数
MAX(e1,e2, ...)	传回e1、e2、……较大数
ABS(expression)	返回非负数expression的绝对值
ROUND(expression,precision)	返回按指定位数进行四舍五入的数值。
FLOOR(expression)	不大于expression的最大整数
CEILING(expression)	不小于expression的最小整数
LOG(expression)	以常数e为底数的对数值
LOG10(expression)	以10为底数的对数值
SQRT(expression)	返回expression的开平方值
SIN(expression)	正弦函数值
COS(expression)	余弦函数值
TAN(expression)	正切函数值
ASIN(expression)	反正弦函数值
ACOS(expression)	反余弦函数值
ATAN(expression)	反正切函数值
SINH(expression)	双曲正弦值
COSH(expression)	双曲余弦值
TANH(expression)	双曲正切值
RAD(expression)	将角度转换为弧度
DEG(expression)	将弧度转换为角度

常量

运算符	描述
e	自然常数e，精确到70位
PI	圆周率，精确到100位
TRUE	数值1
FAUSE	数值0

百度特有函数

运算符	描述
swap16(x)	把原始数据的前后 8 个 bits 互换 (byte1byte2 -> byte2byte1)
swap32(x)	把原始数据的前后 16 个 bits 互换 (byte1byte2byte3 byte4 -> byte3byte4byte1 byte2)
tofloat(x)	按ieee754的标准解析浮点数据，比如 tofloat(x)，当数据是“0x451e1000”是，会返回2529.0

第3章 产品定价

百度云物解析服务为免费服务，实体设备与物解析交互产生的消息量（如设备端数据上传、物解析指令下发等）将计入物接入服务的当月套餐额度。关于物接入的计费标准，请查看[物接入产品定价](#)。

第4章 API参考

4.1 介绍

4.1.1 简介

物解析是在云端为用户提供工业协议解析服务（比如Modbus和OPC UA）服务。当云端收到网关返回的原始数据后，结合用户提供的设备通讯地址表，“物解析”将数据解析成直接可用于存储和分析的数据。

物解析API主要提供解析项目列表的筛选，解析规则的筛选、获取和创建等接口。

4.1.2 调用方式

概述 物解析的设计采用了Restful风格，每个API功能（也可以称之为资源）都使用URI（Universal Resource Identifier）来唯一确定。对资源的请求方式是通过向资源对应的URI发送标准的HTTP请求，比如GET、PUT、POST等，同时，请求需要遵守签名算法，并包含约定的请求参数

通用约定

- 所有编码都采用UTF-8
- 日期格式采用yyyy-MM-dd方式，如2015-08-10
- 时间格式采用UTC格式：yyyy-MM-ddTHH:mm:ssZ, 如2015-08-20T01:24:32Z
- Content-type为application/json; charset=UTF-8
 - object类型的key必须使用双引号（"）括起来
 - object类型的key必须使用lowerCamelCase表示

头域（Header）	是否必须	说明
Authorization	必须	包含Access Key与请求签名
Host	必须	包含API的域名
Content-Type	可选	application/ charset=utf-8

公共请求头

头域 (Header)	说明
Content-Type	只支持JSON格式, application/ json; charset=utf-8
x-bce-request-id	后端生成, 并自动设置到响应头域中

公共响应头

响应状态码 返回的响应状态码遵循[RFC 2616 section 6.1.1](#)

- 1xx: Informational - Request received, continuing process.
- 2xx: Success - The action was successfully received, understood, and accepted.
- 3xx: Redirection - Further action must be taken in order to complete the request.
- 4xx: Client Error - The request contains bad syntax or cannot be fulfilled.
- 5xx: Server Error - The server failed to fulfill an apparently valid request.

通用错误返回格式 当调用接口出错时, 将返回通用的错误格式。Http的返回状态码为4xx或5xx, 返回的消息体将包括全局唯一的请求、错误代码以及错误信息。调用方可根据错误码以及错误信息定位问题, 当无法定位到错误原因时, 可以发工单联系百度技术人员, 并提供requestid以便于快速地帮助您解决问题。

消息体定义

参数名	类型	说明
requestId	String	请求的唯一标识
code	String	错误类型代码
message	String	错误的信息说明

错误返回示例

" requestId" : " 47e0ef1a-9bf2-11e1-9279-0100e8cf109a" ,

" code" : " NoSuchKey" ,

" message" : " The resource you requested does not exist"

```
" requestId" :  
  " 47e0ef1a-9bf2-11e1-9279-0100e8cf109a" , " code" : " NoSuchKey" , " message" : " The  
  resource you requested does not exist"
```

Code	Message	HTTP Status Code	说明
BceValidationException	[param]: [param]=[Validation criteria]	400	无效的[param]参数
MoneyNotEnough	Money not enough to complete the current request	400	余额不足以完成当前的请求操作
SignatureDoesNotMatch	The request signature we calculated does not match the signature you provided. Check your Secret Access Key and signing method. Consult the service documentation for details	400	Authorization 头域中附带的签名和服务端验证不一致
InvalidAccessKeyId	The Access Key ID you provided does not exist in our records	403	Access Key ID不存在
ServiceInternal Error	Service internal error occurred	500	内部服务发生错误

公共错误码

签名认证 物解析API会对每个访问的请求进行身份认证，以保障用户的安全。安全认证采用Access Key与请求签名机制。Access Key由Access Key ID和Secret Access Key组成，均为字符串，由百度云官方颁发给用户。其中Access Key ID用于标识用户身份，Access Key Secret 是用于加密签名字符串和服务器端验证签名字符串的密钥，必须严格保密。

对于每个HTTP请求，用户需要使用下文所描述的方式生成一个签名字符串，并将认证字符串放在HTTP请求的Authorization头域里。

签名字符串格式

bce-auth-v{version}/{accessKeyId}/{timestamp}/{expireTime}/{signedHeaders}/{signature}

其中：

- version是正整数，目前取值为1。
- timestamp是生成签名时的时间。时间格式符合[通用约定](#)。

- expireTime表示签名有效期限，单位为秒，从timestamp所指定的时间开始计算。
- signedHeaders是签名算法中涉及到的头域列表。头域名字之间用分号(;)分隔，如host;x-bce-date。列表按照字典序排列。当signedHeaders为空时表示取默认值。
- signature是256位签名的十六进制表示，由64个小写字母组成，生成方式由如下[签名生成算法](#)给出。

签名生成算法 有关签名生成算法的具体介绍，请参看[鉴权认证机制](#)。

4.1.3 多区域选择

物解析目前仅支持“华南-广州”区域，区域的API地址为：parser.iot.gz.baidubce.com

4.2 API接口描述

4.2.1 headers

说明

- 所有资源如无特殊说明，创建出来的时候都是State=ENABLED，Status=ACTIVE
- 不同的bce user 之间的资源是隔离的

除了no-auth开头的API外，其余的API需要在header里带上authorization的头，如：authorization : /com/baidu/xxx/xxx/xxx/xx/xxx

content-type 都是 application/json

4.2.2 排序 (order)

支持所有list api

参数说明：

名称	**类型**	**是否必选**	**默认值**	**说明**
order	ENUM['desc' , 'asc']	N	desc	排序的方式，不区分大小写
orderBy	String	N	createTime	另外一个支持排序的字段是name

4.2.3 分页 (pagination)

支持所有list api

名称	**类型**	**是否必选**	**默认值**	**说明**
pageNo	Int	N	1	页码
pageSize	Int	N	50	每页item个数

4.2.4 搜索 (search)

支持所有list api

名称	**类型**	**是否必选**	**默认值**	**说明**
q	String	N	""	模糊查询的内容。目前所有list都忽略这个字段（没有需求）

4.3 Gateway

字段	**类型**	**说明**
uuid	String	全局唯一ID
code	String	网关名称。同一个bce用户下唯一
description	String	网关描述
principalUuid	String	网关对应设备的身份
policyUuid	String	网关对应设备的身份对应的策略
useSsl	Boolean	网关是否使用SSL通信
host	String	数据下发和上传的 endpoint。当useSsl=true时是tls地址，否则为tcp地址
username	String	网关和Mqtt交互所需的username
password	String	网关和Mqtt交互所需的password

字段	**类型**	**说明**
commandTopic	String	配置下发的主题
state	ENUM	[ENABLED, DISABLED]
status	ENUM	[ERROR, ACTIVE, DELETED]
createTime	DateTime	创建记录的时间
updateTime	DateTime	最后修改记录的时间

4.3.1 分页查询

接口描述

相对URI	**HTTP 方式**
/v1/gateway	GET

输入

名称	**类型**	**是否必选**	**默认值**	**含义**
state	ENUM	N	null	筛选条件之一
status	ENUM	N	null	筛选条件之一

输出

200: 成功

输出一个Gateway的List

其他: 失败

输出示例

```
{
  "totalCount": 2,
  "result": [
    {
      "username": "parser\\_endpoint1482460638850/mb\\_thing\\_v21486196129170",
      "policyUuid": "mb\\_policy\\_v21486196129170",
      "useSsl": false,
      "state": "ENABLED",
      "description": "jwan test bos",
      "code": "jwangateway",
      "host":
        "[tcp://abc.mqtt.gz.baidubce.com:1883](tcp://abc.mqtt.gz.baidubce.com:1883)",
      "password": "abcafdadsfadf",
    }
  ]
}
```



```

" principalUuid" : " mb\\_principal\\_v21486196129170" ,
" createTime" : " 2017-02-04T08:15:29Z" ,
" status" : " ACTIVE" ,
" updateTime" : " 2017-02-04T08:15:29Z" ,
" commandTopic" : " mb\\_commandTopic\\_v21486196129170" ,
" uuid" : " 08d95c37-bc72-44cd-8e93-8d94ff52a934"
},
{
" username" : " parser\\_endpoint1482460638850/mb\\_thing\\_v21484554172501" ,
" policyUuid" : " mb\\_policy\\_v21484554172501" ,
" useSsl" : true,
" state" : " ENABLED" ,
" description" : " cc" ,
" code" : " testCaseGateway" ,
" host" :
" [ssl://abc.mqtt.gz.baidubce.com:1884](ssl://abc.mqtt.gz.baidubce.com:1884)" ,
" password" : " dfdf" ,
" principalUuid" : " mb\\_principal\\_v21484554172501" ,
" createTime" : " 2017-01-16T08:09:33Z" ,
" status" : " ACTIVE" ,
" updateTime" : " 2017-01-16T08:09:33Z" ,
" commandTopic" : " mb\\_commandTopic\\_v21484554172501" ,
" uuid" : " ef70bff4-535c-44d3-8293-3a123fa36537"
}
],
" order" : " desc" ,
" orderBy" : " createTime" ,
" pageSize" : 10,
" pageNo" : 1
}

```

4.3.2 获取

接口描述

相对URI	**HTTP 方式**
/v1/gateway/:gatewayUuid	GET

输入

名称	**类型**	**是否必选**	**默认值**	**含义**
gatewayUuid	String	Y		网关Id

输出

200: 成功

输出一个这个uuid对应的gateway

其他: 失败

输出示例

```
{
  "username": "parser\\_endpoint1472628546412/mb\\_thing\\_v21481002980078",
  "policyUuid": "mb\\_policy\\_v21481002980078",
  "useSsl": false,
  "state": "ENABLED",
  "description": "created by api",
  "code": "yyjGateway08",
  "host": "[tcp://10.73.203.34:8061](tcp://10.73.203.34:8061)",
  "password": "vV/ldfasdfadFig=",
  "principalUuid": "mb\\_principal\\_v21481002980078",
  "createTime": "2016-12-06T05:43:02Z",
  "status": "ACTIVE",
  "updateTime": "2016-12-06T05:43:02Z",
  "commandTopic": "mb\\_commandTopic\\_v21481002980078",
  "uuid": "4ac62c7e-da98-4c1e-95df-2313cf66754b"
}
```

4.3.3 创建

接口描述

相对URI	**HTTP 方式**
/v1/gateway	POST

输入

名称	**类型**	**是否必选**	**默认值**	**参数位置**	**说明**
code	String	Y		body json	网关名称
description	String	N	""	body json	网关描述
useSsl	Boolean	Y		body json	是否使用SSL通信

输出

201: 成功

输出创建成功的gateway

其他: 失败

输出示例

```
{
  "username": "parser\\_endpoint1472628546412/mb\\_thing\\_v21481002980078",
  "policyUuid": "mb\\_policy\\_v21481002980078",
  "useSsl": false,
  "state": "ENABLED",
  "description": "created by api",
  "code": "yyjGateway08",
  "host": "[tcp://10.73.203.34:8061](tcp://10.73.203.34:8061)",
  "password": "vW/ldfasdfadFig=",
  "principalUuid": "mb\\_principal\\_v21481002980078",
  "createTime": "2016-12-06T05:43:02Z",
  "status": "ACTIVE",
  "updateTime": "2016-12-06T05:43:02Z",
  "commandTopic": "mb\\_commandTopic\\_v21481002980078",
  "uuid": "4ac62c7e-da98-4c1e-95df-2313cf66754b"
}
```

4.3.4 更新

接口描述

相对URI	**HTTP 方式**
/v1/gateway/:gatewayUuid	PUT

输入

不能所有字段都为空

名称	**类型**	**是否必选**	**默认值**	**参数位置**	**说明**
description	String	N		body json	网关描述
useSsl	Boolean	N		body json	是否使用SSL通信
state	ENUM	N		body json	状态, [ENABLED, DISABLED]

输出

201: 成功

输出更新成功后的gateway

其他: 失败

输出示例

```
{
  "username": "parser\\_endpoint1472628546412/mb\\_thing\\_v21481002980078",
  "policyUuid": "mb\\_policy\\_v21481002980078",
  "useSsl": false,
  "state": "ENABLED",
  "description": "created by api",
  "code": "yyjGateway08",
  "host": "[tcp://10.73.203.34:8061](tcp://10.73.203.34:8061)",
  "password": "vV/ihR90Zp/HLBwo3b0fHVI14PNVejTI3Cp7e2AVFig=",
  "principalUuid": "mb\\_principal\\_v21481002980078",
  "createTime": "2016-12-06T05:43:02Z",
  "status": "ACTIVE",
  "updateTime": "2016-12-06T05:43:02Z",
  "commandTopic": "mb\\_commandTopic\\_v21481002980078",
  "uuid": "4ac62c7e-da98-4c1e-95df-2313cf66754b"
}
```

4.3.5 重置密钥

接口描述

相对URI	**HTTP 方式**
/v1/gateway/:gatewayUuid/regen	PUT

输出

201: 成功

输出更新成功后的gateway

其他: 失败

输出示例

```
{
  "username": "parser\\_endpoint1472628546412/mb\\_thing\\_v21481002980078",
  "policyUuid": "mb\\_policy\\_v21481002980078",
  "useSsl": false,
  "state": "ENABLED",
  "description": "created by api",
  "code": "yyjGateway08",
}
```

```

" host" : " [tcp://10.73.203.34:8061](tcp://10.73.203.34:8061)" ,
" password" : " vV/iHR90Zp/HLBwo3b0fHVI14PNVejTI3Cp7e2AVFig=" ,
" principalUuid" : " mb\\_principal\\_v21481002980078" ,
" createTime" : " 2016-12-06T05:43:02Z" ,
" status" : " ACTIVE" ,
" updateTime" : " 2016-12-06T05:43:02Z" ,
" commandTopic" : " mb\\_commandTopic\\_v21481002980078" ,
" uuid" : " 4ac62c7e-da98-4c1e-95df-2313cf66754b"
}

```

4.3.6 删除

接口描述

相对URI	**HTTP 方式**
/v1/gateway/:gatewayUuid	DELETE

输入

名称	**类型**	**是否必选**	**默认值**	**含义**
gatewayUuid	String	Y		网关id

输出

204: 成功

其他: 失败

4.4 Device

字段	**类型**	**说明**
uuid	String	全局唯一ID
code	String	设备名称。同一个gateway下唯一
description	String	设备描述
gatewayUuid	String	网关id
gatewayCode	String	冗余字段。与gatewayUuid对应的gateway的code字段保持一致

字段	类型	说明
slaveId	Int	用于 modbus。只能从 1-247 (包括1和237) 中取值且同一个gateway下唯一
address	String	用于子设备的ip:port, 或者串口地址(eg: /dev/pts/6)
mode	ENUM	[TCP/RTU]。默认值TCP
baud	Int	波特率, 仅用于RTU模式
databits	Int	数据位长, [7-8], 默认值8, 仅用于RTU模式
stopbits	Int	停止位长, [1,2], 默认值1, 仅用于RTU模式
parity	ENUM	校验方式, [NONE/EVEN/ODD], 默认值NONE, 仅用于RTU模式
state	ENUM	[ENABLED, DISABLED]
status	ENUM	[ERROR, ACTIVE, DELETED]
createTime	DateTime	创建记录的时间
updateTime	DateTime	最后修改记录的时间

4.4.1 筛选

接口描述

相对URI	HTTP 方式
/v1/device	GET

输入

名称	类型	是否必选	默认值	含义
gatewayUuid	String	N	null	网关id
state	ENUM	N	null	[ENABLED, DISABLED]

输出

200: 成功

输出一个Device的List

其他: 失败

输出示例

```
{
  "totalCount": 1,
  "result": [
    {
      "databits": 8,
      "baud": 300,
      "address": "127.0.0.1:502",
      "gatewayUuid": "08d95c37-bc72-44cd-8e93-8d94ff52a934",
      "state": "ENABLED",
      "description": "",
      "code": "subdevice",
      "mode": "TCP",
      "createTime": "2017-02-04T08:16:11Z",
      "parity": "NONE",
      "status": "ACTIVE",
      "slaveId": 1,
      "stopbits": 1,
      "updateTime": "2017-02-04T08:16:11Z",
      "gatewayCode": "jwangateway",
      "uuid": "8603dbc2-3fdc-4683-bfc5-dc18ed50adcf"
    }
  ],
  "order": "desc",
  "orderBy": "createTime",
  "pageSize": 5,
  "pageNo": 1
}
```

4.4.2 获取

接口描述

相对URI	**HTTP 方式**
/v1/device/:deviceUuid	GET

输入

名称	**类型**	**是否必选**	**默认值**	**含义**
deviceUuid	String	Y		子设备id

输出

200: 成功

输出一个这个uuid对应的device

其他: 失败

输出示例

```
{
  " databits" : 8,
  " baud" : 300,
  " address" : " 127.0.0.1:502" ,
  " gatewayUuid" : " 08d95c37-bc72-44cd-8e93-8d94ff52a934" ,
  " state" : " ENABLED" ,
  " description" : " " ,
  " code" : " subdevice" ,
  " mode" : " TCP" ,
  " createTime" : " 2017-02-04T08:16:11Z" ,
  " parity" : " NONE" ,
  " status" : " ACTIVE" ,
  " slaveId" : 1,
  " stopbits" : 1,
  " updateTime" : " 2017-02-04T08:16:11Z" ,
  " gatewayCode" : " jwangateway" ,
  " uuid" : " 8603dbc2-3fdc-4683-bfc5-dc18ed50adcf"
}
```

4.4.3 创建

接口描述

相对URI	**HTTP 方式**
/v1/device	POST

输入

名称	**类型**	**是否必选**	**默认值**	**参数位置**	**说明**
code	String	Y		body json	子设备名称 (同一网关下唯一)
description	String	N	""	body json	子设备描述
gatewayUuid	String	Y		body json	所属网关id
slaveId	Int	Y		body json	MODBUS 从站号[1-247]

名称	**类型**	**是否必选**	**默认值**	**参数位置**	**说明**
address	String	Y		body json	地址，对于 TCP 模式：ip:port, 如 192.168.0.2:502 对于 RTU 模式：是串口号，如/dev/USBtty1
mode	ENUM	N	TCP	body json	TCP, RTU
baud	Int	N	9600	body json	波特率，仅对 RTU 有意义
databits	Int	N	8	body json	数据位，仅对 RTU 有意义
stopbits	Int	N	1	body json	停止位，仅对 RTU 有意义
parity	ENUM	N	NONE	body json	校验，NONE/ EVEN/ ODD, 仅对 RTU 有意义

输出

201: 成功

输出创建成功的device

其他: 失败

输出示例

```
{
  " databits" : 8,
  " baud" : 300,
  " address" : " 127.0.0.1:502" ,
  " gatewayUuid" : " 08d95c37-bc72-44cd-8e93-8d94ff52a934" ,
  " state" : " ENABLED" ,
  " description" : " " ,
  " code" : " subdevice" ,
  " mode" : " TCP" ,
```

```

" createTime" : " 2017-02-04T08:16:11Z" ,
" parity" : " NONE" ,
" status" : " ACTIVE" ,
" slaveId" : 1,
" stopbits" : 1,
" updateTime" : " 2017-02-04T08:16:11Z" ,
" gatewayCode" : " jwangateway" ,
" uuid" : " 8603dbc2-3fdc-4683-bfc5-dc18ed50adcf"
}

```

4.4.4 更新

接口描述

相对URI	**HTTP 方式**
/v1/device/:deviceUuid	PUT

输入

不能所有字段都为空

名称	**类型**	**是否必选**	**默认值**	**参数位置**	**说明**
description	String	N		body json	描述
slaveId	Int	N		body json	modbus 从站号
address	String	N		body json	地址
state	ENUM	N		body json	状态, [ENABLED, DISABLED]
mode	ENUM	N		body json	TCP,RTU
baud	Int	N		body json	波特率
databits	Int	N		body json	数据位
stopbits	Int	N		body json	停止位
parity	ENUM	N		body json	奇偶校验

输出

201: 成功

输出更新成功后的device

其他: 失败

4.4.5 删除

接口描述

相对URI	**HTTP 方式**
/v1/device/:deviceUuid	DELETE

输入

名称	**类型**	**是否必选**	**默认值**	**含义**
deviceUuid	String	Y		子设备id

输出

204: 成功

其他: 失败

4.5 ParserObject

字段	**类型**	**说明**
uuid	String	全局唯一ID
name	String	项目名称
description	String	项目描述
gatewayUuid	String	对应网关id
gatewayCode	String	冗余字段。与gatewayUuid对应的gateway的code字段保持一致
storage	String	BOS上的地址, [可以为空]
storageTaskId	String	转储数据到BOS的任务名称
protocol	ENUM	[MODBUS]目前只能填这一个, 也是默认值
dataTopic	String	网关上传数据使用的MQTT主题

字段	**类型**	**说明**
state	ENUM	[ERROR, PAUSED, RUNNING]
status	ENUM	[ERROR, ACTIVE, DELETED]
createTime	DateTime	创建记录的时间
updateTime	DateTime	最后修改记录的时间
destTopic	String	解析后的消息写入那个mqtt topic, [可以为空]

筛选 接口描述

相对URI	**HTTP 方式**
/v1/parser-object	GET

输入

名称	**类型**	**是否必选**	**默认值**	**含义**
gatewayUuid	String	N	null	网关id
state	ENUM	N	null	‘RUNNING’ 或 者 ‘PAUSED’

输出

200: 成功

输出一个parserObject的List

其他: 失败

输出示例:

```
{
  "orderBy": " createTime",
  "order": " desc",
  "pageNo": 1,
  "pageSize": 10,
  "result": [
    {
      "uuid": " 3568afb6-31a1-4686-80a5-620b7670659b",
      "state": " RUNNING",
      "dataTopic": " mb\\_dataTopic\\_v21476863385409",
      "name": " ldwde2",
```

```
" protocol" : " MODBUS" ,
" status" : " ACTIVE" ,
" gatewayUuid" : " f723e888-0f86-4992-a626-2c5c081e93f6" ,
" gatewayCode" : " ldwgate" ,
" storage" : " " ,
" storageTaskId" : " mb\\_storage\\_v21476863385409" ,
" createTime" : " 2016-10-19T07:49:46Z" ,
" updateTime" : " 2016-10-19T07:49:46Z" ,
" destTopic" : " ldwde2"
},
{
" uuid" : " cc49c9b8-22cc-4ffe-929e-d4da99a37438" ,
" state" : " RUNNING" ,
" dataTopic" : " mb\\_dataTopic\\_v21476852302035" ,
" name" : " ldwtestv22" ,
" protocol" : " MODBUS" ,
" status" : " ACTIVE" ,
" gatewayUuid" : " f723e888-0f86-4992-a626-2c5c081e93f6" ,
" gatewayCode" : " ldwgate" ,
" storage" : " [bos://ldwldw-test/](bos://ldwldw-test/)" ,
" storageTaskId" : " mb\\_storage\\_v21476852302035" ,
" createTime" : " 2016-10-19T04:45:02Z" ,
" updateTime" : " 2016-10-19T04:45:02Z" ,
" destTopic" : " ldwtopic"
}
],
" totalCount" : 3
}
```

4.5.1 获取

接口描述

相对URI	**HTTP 方式**
/v1/parser-object/:parserObjectUuid	GET

输入

名称	**类型**	**是否必选**	**默认值**	**含义**
parserObjectUuid	String	Y		解析项目id

输出

200: 成功

输出一个这个uuid对应的parserObject

其他: 失败

输出示例:

```
{
  " uuid" : " cc49c9b8-22cc-4ffe-929e-d4da99a37438" ,
  " state" : " RUNNING" ,
  " dataTopic" : " mb\\_dataTopic\\_v21476852302035" ,
  " name" : " ldwtestv22" ,
  " protocol" : " MODBUS" ,
  " status" : " ACTIVE" ,
  " gatewayUuid" : " f723e888-0f86-4992-a626-2c5c081e93f6" ,
  " gatewayCode" : " ldwgate" ,
  " storage" : " [bos://ldwldw-test/](bos://ldwldw-test/)" ,
  " storageTaskId" : " mb\\_storage\\_v21476852302035" ,
  " createTime" : " 2016-10-19T04:45:02Z" ,
  " updateTime" : " 2016-10-19T04:45:02Z" ,
  " destTopic" : " ldwtopic"
}
```

4.5.2 创建

接口描述

相对URI	**HTTP 方式**
/v1/parser-object	POST

输入

名称	**类型**	**是否必选**	**默认值**	**参数位置**	**说明**
name	String	Y		body json	名称
gatewayUuid	String	Y		body json	网关id
storage	String	N		body json	存 储 到 bos 的地址
protocol	ENUM	N	MODBUS	body json	只 能 是 MODBUS
destTopic	String	N	NULL		转发到mqtt 目的地的主题

输出

201: 成功

输出创建成功的parserObject

其他: 失败

输出示例:

```
{
  " uuid" : " cc49c9b8-22cc-4ffe-929e-d4da99a37438" ,
  " state" : " RUNNING" ,
  " dataTopic" : " mb\\_dataTopic\\_v21476852302035" ,
  " name" : " ldwtestv22" ,
  " protocol" : " MODBUS" ,
  " status" : " ACTIVE" ,
  " gatewayUuid" : " f723e888-0f86-4992-a626-2c5c081e93f6" ,
  " gatewayCode" : " ldwgate" ,
  " storage" : " [bos://ldwldw-test/](bos://ldwldw-test/)" ,
  " storageTaskId" : " mb\\_storage\\_v21476852302035" ,
  " createTime" : " 2016-10-19T04:45:02Z" ,
  " updateTime" : " 2016-10-19T04:45:02Z" ,
  " destTopic" : " ldwtopic"
}
```

4.5.3 更新

接口描述

相对URI	**HTTP 方式**
/v1/parser-object/<parserObjectUuid>	PUT

输入

名称	**类型**	**是否必选**	**默认值**	**参数位置**	**说明**
state	ENUM	N			‘RUNNING’ 或 者 ‘PAUSED’
name	String	N		body json	名称
gatewayUuid	String	N		body json	网关id
storage	String	N		body json	存 储 到 bos 的地址

名称	**类型**	**是否必选**	**默认值**	**参数位置**	**说明**
protocol	ENUM	N		body json	只能是MODBUS
destTopic	String	N			转发到mqtt目的地的主题

输出

201: 成功

输出更新成功后的parserObject

其他: 失败

输出示例:

```
{
  " destTopic" : " destTopicYyj6b" ,
  " gatewayUuid" : " 9d00d091-b37e-4a91-8562-89b14826e687" ,
  " state" : " RUNNING" ,
  " dataTopic" : " mb\\_dataTopic\\_v31480492111054" ,
  " name" : " yyjPrj6b" ,
  " createTime" : " 2016-11-30T07:48:34Z" ,
  " protocol" : " MODBUS" ,
  " status" : " ACTIVE" ,
  " updateTime" : " 2016-11-30T07:48:34Z" ,
  " gatewayCode" : " yyjgateway" ,
  " storageTaskId" : " mb\\_storage\\_v31480492111054" ,
  " uuid" : " 061e26ab-5f8f-421b-9a96-ce584fc59eb0" ,
  " storage" : " [bos://ldwldw-test/sub/test6b](bos://ldwldw-test/sub/test6b)"
}
```

4.5.4 删除

接口描述

相对URI	**HTTP 方式**
/v1/parser-object/<parserObjectUuid>	DELETE

输入

名称	**类型**	**是否必选**	**默认值**	**含义**
parserObjectUuid	String	Y		解析项目id

输出

204: 成功

其他: 失败

4.6 PullRule

字段	**类型**	**说明**
uuid	String	全局唯一ID
parserObjectUuid	String	解析项目id
deviceUuid	String	子设备的id
functionCode	Int	操作码: 1,2,3,4
startAddress	Int	起始地址[0, 65535]
length	Int	请求数据长度[1, 65536]
pullInterval	Int	轮询间隔(秒)
status	ENUM	[ERROR, ACTIVE, DELETED]
createTime	DateTime	创建记录的时间
updateTime	DateTime	最后修改记录的时间

4.6.1 筛选

接口描述

相对URI	**HTTP 方式**
/v1/pull-rule	GET

输入

名称	**类型**	**是否必选**	**默认值**	**含义**
parserObjectUuid	String	N	null	筛选条件之一，解析项目id

名称	**类型**	**是否必选**	**默认值**	**含义**
deviceId	String	N	null	筛选条件之一，子设备id
status	ENUM	N	null	筛选条件之一，状态
withDevice	Boolean	N	false	如果添加会多返回一个device字段 “device”：{ “slaveId”：Int, “code”：String, “address”：String, “description”：String }

输出

200: 成功

输出一个Device的List

其他: 失败

输出示例

```
{
  "totalCount": 1,
  "result": [
    {
      "functionCode": 3,
      "length": 3,
      "createTime": "2017-03-07T17:05:40Z",
      "status": "ACTIVE",
      "parserObjectUuid": "cc6904fd-d6ef-4bbb-9b78-36b243627cb7",
      "updateTime": "2017-03-07T17:05:40Z",
      "pullInterval": 1,
      "startAddress": 9,
      "deviceId": "d6cf5a60-cc0d-45b6-bb4c-3a2a1829279e",
      "uuid": "0feb2135-2646-4519-86c8-ca88e412e0b7"
    }
  ],
  "order": "asc",
  "orderBy": "createTime",
  "pageSize": 50,
  "pageNo": 1
}
```

```
}

```

4.6.2 获取

接口描述

相对URI	**HTTP 方式**
/v1/pull-rule/: pullRuleUuid	GET

输入

名称	**类型**	**是否必选**	**默认值**	**含义**
withDevice	Boolean	N	false	如果添加会多返回一个device字段 “device” : { “slaveId” : Int, “code” : String, “ad- dress” : String , “descrip- tion” : String }
pullRuleUuid	String	Y		轮询规则id

输出

200: 成功

输出一个这个uuid对应的pullrule

其他: 失败

输出示例

```
{
  " functionCode" : 3,
  " length" : 8,
  " device" : {
    " slaveId" : 218,
    " code" : " cbaf6cf7-896c-4a76-9f77-2797c2db1d00\_1488906401944" ,
    " address" : " /dev/TTYUsb1" ,
    " description" : " device for cbaf6cf7-896c-4a76-9f77-2797c2db1d00"
  },
  " createTime" : " 2017-03-07T17:06:42Z" ,

```

```

" status" : " ACTIVE" ,
" parserObjectUuid" : " c39bc78b-5093-41bb-b1a4-9ee3cae47ca2" ,
" updateTime" : " 2017-03-07T17:06:42Z" ,
" pullInterval" : 5,
" startAddress" : 5,
" deviceUuid" : " 6ebafd28-a078-44aa-aae3-ebb9774f127a" ,
" uuid" : " 766c1880-bfd4-449a-9304-84192ae5851e"
}

```

4.6.3 创建

接口描述

相对URI	**HTTP 方式**
/v1/pull-rule	POST

输入

名称	**类型**	**是否必选**	**默认值**	**参数位置**	**说明**
parserObjectUuid	String	Y		body json	解析项目id
deviceUuids	List[String]	Y		body json	子设备id列表
fucntionCode	Int	Y		body json	操作码 [1,2,3,4]
startAddress	Int	Y		body json	起始地址
length	Int	Y		body json	请求数据长度
pullInterval	Int	Y		body json	轮询间隔 (秒)

输出

201: 成功

输出创建成功的pull rule

其他: 失败

输出示例

```

{
" parserObjectUuid" : " 227d290c-ddd1-446d-a9e3-e4841afe8535" ,

```

```
" deviceUids" : [
" 842b91c9-f66f-455b-a57e-5ddc4dc2edc2"
],
" functionCode" : 1,
" startAddress" : 27,
" length" : 1,
" pullInterval" : 3
}
```

4.6.4 更新

接口描述

相对URI	**HTTP 方式**
/v1/pull-rule/:pullRuleUuid	PUT

输入

不能所有字段都为空

名称	**类型**	**是否必选**	**默认值**	**参数位置**	**说明**
pullInterval	Int	N		body json	轮询间隔

输出

201:
输出修改成功后的pull rule
其他: 失败

4.6.5 删除

接口描述

相对URI	**HTTP 方式**
/v1/pull-rule/:pullRuleUuid	DELETE

输入

名称	**类型**	**是否必选**	**默认值**	**含义**
pullRuleUuid	String	Y		轮询规则id

输出

204: 成功

其他: 失败

4.7 DataDescription

| **字段** | **类型** | **说明** | |-----|-----|-----|
 |-----| | uuid | String | 全局唯一ID | | parserObjectUuid | String
 | 解析项目id | | bit | String | 不可以>=length | | length | ENUM | [1,8,16,32] | | address
 | Int | 对应的MODBUS设备地址，例如40001，00002 | | name | String | 名称 | | kind |
 ENUM | [INT, BOOL, REAL, INT32, REAL32] | | unit | String | 单位 | | rh | Double | 上限
 | | rl | Double | 下限 | | formula | String | 计算公式（通过x引用modbus数据） | | user
 _properties | JSON | 用户自定义属性，以JSON格式标示, eg: { “myprop” : “thisisvalue” ,
 “myprop1” : “val2” } | | state | ENUM | [ENABLED, DISABLED] | | status | ENUM |
 [ERROR, ACTIVE, DELETED] | | createTime | DateTime | 创建记录的时间 | | updateTime |
 DateTime | 最后修改记录的时间 |

4.7.1 筛选

接口描述

相对URI	**HTTP 方式**
/v1/data-description	GET

输入

名称	**类型**	**是否必选**	**默认值**	**含义**
parserObjectUuid	String	N	null	解析项目id
state	ENUM	N	null	[ENABLED, DISABLED]
addressStart	Int	N	\-1	address以什么开头。可能取值:-1,0,1,3,4。（-1和不填一样）

输出

200: 成功

输出一个data description的List

其他: 失败

示例输出

```
{
  "orderBy": " createTime" ,
  "order": " desc" ,
  "pageNo": 1,
  "pageSize": 10,
  "result": [
    {
      "formula": " x+4" ,
      "user\\_properties": {},
      "rl": 0,
      "address": " 40004" ,
      "abbreviation": " " ,
      "state": " ENABLED" ,
      "kind": " INT" ,
      "length": 16,
      "rh": 1,
      "bit": -1,
      "name": " overwrite" ,
      "createTime": " 2016-11-17T07:37:32Z" ,
      "status": " ACTIVE" ,
      "parserObjectUuid": " b838bb59-8f5a-4e2a-aa1e-a3694425c46c" ,
      "updateTime": " 2016-11-17T07:37:32Z" ,
      "unit": " a" ,
      "uuid": " 7ea063dd-5fca-4fad-b38b-726066eca276"
    },
    {
      "formula": " " ,
      "user\\_properties": {},
      "rl": 0,
      "address": " 40003" ,
      "abbreviation": " " ,
      "state": " ENABLED" ,
      "kind": " INT" ,
      "length": 16,
      "rh": 1,
      "bit": -1,
      "name": " asdfa" ,
      "createTime": " 2016-11-17T01:50:32Z" ,
      "status": " ACTIVE" ,
      "parserObjectUuid": " b838bb59-8f5a-4e2a-aa1e-a3694425c46c" ,
      "updateTime": " 2016-11-17T01:50:32Z" ,
```

```
" unit" : " " ,
" uuid" : " df68b7dd-7920-4cbd-a9c2-57c19642c278"
}
],
" totalCount" : 2
}
```

4.7.2 获取

接口描述

相对URI	**HTTP 方式**
/ v1/ data-description/ <data-description-uuid>	GET

输入

名称	**类型**	**是否必选**	**默认值**	**含义**
dataDescription Uuid	String	Y		解析设置id

输出

200: 成功

输出一个这个uuid对应的data description

其他: 失败

示例输出

```
{
" formula" : " x+4" ,
" user\\_properties" : {},
" r1" : 0,
" address" : " 40004" ,
" abbreviation" : " " ,
" state" : " ENABLED" ,
" kind" : " INT" ,
" length" : 16,
" rh" : 1,
" bit" : -1,
" name" : " overwrite" ,
" createTime" : " 2016-11-17T07:37:32Z" ,
" status" : " ACTIVE" ,
" parserObjectUuid" : " b838bb59-8f5a-4e2a-aa1e-a3694425c46c" ,
```



```

" updateTime" : " 2016-11-17T07:37:32Z" ,
" unit" : " a" ,
" uuid" : " 7ea063dd-5fca-4fad-b38b-726066eca276"
}

```

4.7.3 创建

接口描述

相对URI	**HTTP 方式**
/v1/data-description	POST

输入

名称	**类型**	**是否必选**	**默认值**	**参数位置**	**说明**
parserObjectUuid	String	Y		body json	解析项目id
bit	int	N	\-1	body json	不能大于length
length	ENUM	Y		body json	数据长度
address	Int	Y		body json	modbus地址
name	String	Y		body json	数据名称
kind	ENUM	Y		body json	[INT, BOOL, REAL, INT32, REAL32
unit	ENUM	N	""	body json	单位
rh	Double	N	1.0	body json	量程上限
rl	Double	N	0.0	body json	量程下限
formula	String	N	""	body json	计算公式
user_properties	JSON				自定义字段

输入示例

```

{
  " address" : " 40001" ,
  " bit" : " " ,
  " name" : " testname" ,
  " rh" : " " ,
  " rl" : " " ,
  " kind" : " INT" ,
  " length" : 16,
  " unit" : " " ,
  " formula" : " x\\*100" ,
  " parserObjectUuid" : " 1ca804a8-80ce-4882-846e-ba8dc733136a" ,
  " user\\_properties" : {分组
  " " : "温度" " ,
  " prop2" : " "
}
}

```

输出

201: 成功

输出创建成功的数据描述

其他: 失败

4.7.4 更新

接口描述

相对URI	**HTTP 方式**
/ v1/ data-description/ <data-description-uuid>	PUT

输入

不能所有字段都为空

| **名称** | **类型** | **是否必选** | **默认值** | **参数位置** | **说明** | |——
———|———|———|———|———|———| | state | ENUM |
N | | body json | [ENABLED, DISABLED] | | bit | int | | | 数据位 | | length | ENUM | |
| | 长度 | | address | int | | | MODBUS数据地址 | | name | String | | | 名字 | | kind
| ENUM | | | 数据类型 | | unit | String | | “” | | 单位 | | rh | Double | | 1 | | 量程
上限 | | rl | Double | | 0 | | 量程下限 | | formula | String | | “” | | 计算公式 | | user
_properties | JSON | | | 自定义属性 |

输入示例

```
{
  " address" : " 40001" ,
  " bit" : " " ,
  " name" : " testname" ,
  " rh" : " " ,
  " rl" : " " ,
  " kind" : " INT" ,
  " length" : 16,
  " unit" : " " ,
  " formula" : " x\\*100" ,
  " parserObjectUuid" : " 1ca804a8-80ce-4882-846e-ba8dc733136a" ,
  " user\\_properties" : {分组
    " " : "温度" " ,
    " prop2" : " "
  }
}
```

输出

201: 成功

输出创建成功的数据描述

其他: 失败

4.7.5 删除

接口描述

相对URI	**HTTP 方式**
/ v1/ data-description/ <data-description-uuid>	DELETE

输入

名称	**类型**	**是否必选**	**默认值**	**含义**
dataDescription Uuid	String	Y		解析设置id

输出

204: 成功

其他: 失败

4.8 Action

4.8.1 下发配置

接口描述

将所有资源的配置下发到gateway上。

如果gateway的状态是DISABLED，就会下发一个空list

如果gateway的状态是ENABLED，就会按照网关的需求把其下所有的device，parser，object等配置项。

相对URI	**HTTP 方式**
/v1/action/deploy-config	POST

输出

201: 成功

返回一个以下数据结构的list

名称	**类型**	**说明**
gatewayUuid	String	所下发的网关的uuid
gatewayCode	String	所下发的网关的代码
result	String	null表示下发成功，否则是异常的说明字符串

其他: 失败

4.9 Custom Property

4.9.1 筛选

接口描述

相对URI	**HTTP 方式**
/v1/custom-field	GET

输出

```
{
  " fields" :[
    " prop1" ,
    " prop2" ,
    " prop3"
  ]
}
```

4.9.2 添加

接口描述

相对URI	**HTTP 方式**
/v1/custom-field	POST

输入

名称	**类型**	**是否必选**	**默认值**	**参数位置**	**说明**
name	String	Y		body json	自定义属性名称

输出

```
{  
  "sucess": "true"  
}
```

4.9.3 删除

接口描述

相对URI	**HTTP 方式**
/v1/custom-field/delete	PUT

输入

名称	**类型**	**是否必选**	**默认值**	**参数位置**	**说明**
name	String	Y		body json	自定义属性名称

输出

```
{  
  "sucess": "true"  
}
```

第5章 常见问题

5.1 物解析常见问题

5.1.1 物解析和物接入的关系是什么？

物解析底层依赖于物接入。物解析是在物接入的基础上，自动创建了一些配置。比如在物解析中新建第一个网关时，系统会自动在物接入中新建一个实例，同时在这个实例中配置设备（thing）、身份（principal）、策略（policy）、主题（topic）等。物解析服务端和网关设备之间，通过这个实例来发布消息或订阅消息。物解析使用的实例在物接入中以 `parser_endpointxxxx` 命名，系统创建的用于设备-身份-策略-主题对用户隐藏。

5.1.2 对网关配置的轮询策略是什么时候写入网关的？

在网关设备管理页面和解析项目页面点击“全部生效”，都会将配置的网关轮询信息下发至网关。

5.1.3 点击“全部生效”时，写入网关的轮询策略是什么？

“在线”状态的网关，物解析服务会将配置好的轮询策略写入网关；“禁用”状态的网关，物解析服务会将空的轮询策略写入网关。

5.1.4 为什么不能新建轮询设置？

请检查网关和网关子设备是否都已创建。

5.1.5 暂停或启动某个解析项目，云端和网关端是如何工作的？

- 暂停解析项目并且点击“全部生效”，会将此命令下发至网关，这个轮询项目对应的轮询规则会被网关忽略，因此，网关端不会采集信息，云端自然不会解析。
- 暂停解析项目但是没有点击“全部生效”，网关会继续按照原有的轮询规则采集数据，但上传后不解析，原始数据直接写入BOS。

- 启动解析项目并且点击“全部生效”，会将此命令下发至网关，网关按照新的轮询策略采集数据到云端，云端按照此解析项目来解析数据。
- 启动解析项目但是没有点击“全部生效”，新的轮询策略并没有下发至网关，网关仍采用上一个轮询策略，可能会与预想的不一致。所以启动解析项目后，需要点击“全部生效”，确保新的轮询策略下发至网关。

5.1.6 禁用或恢复某个网关，云端和网关端是如何工作的？

- 将网关置为禁用或在线状态，并且点击“全部生效”，云端会给置为“禁用”状态的网关写一个空的轮询策略，则此网关不采集数据；给置为“在线”状态的网关写入配置好的轮询策略。
- 将网关置为禁用或在线状态，没有点击“全部生效”，云端不会下发指令给网关。