

第二章 规则链应用设计

规则引擎是物联网平台的一个重要功能模块，主要对从感知层搜集来的设备数据进行过滤、属性（集）、变换、动作、联动、与外部系统交互操作等，实现将业务决策从应用程序代码中分离出来，接收数据输入，解释业务规则，并根据业务规则做出业务决策，如图 2-1 所示。

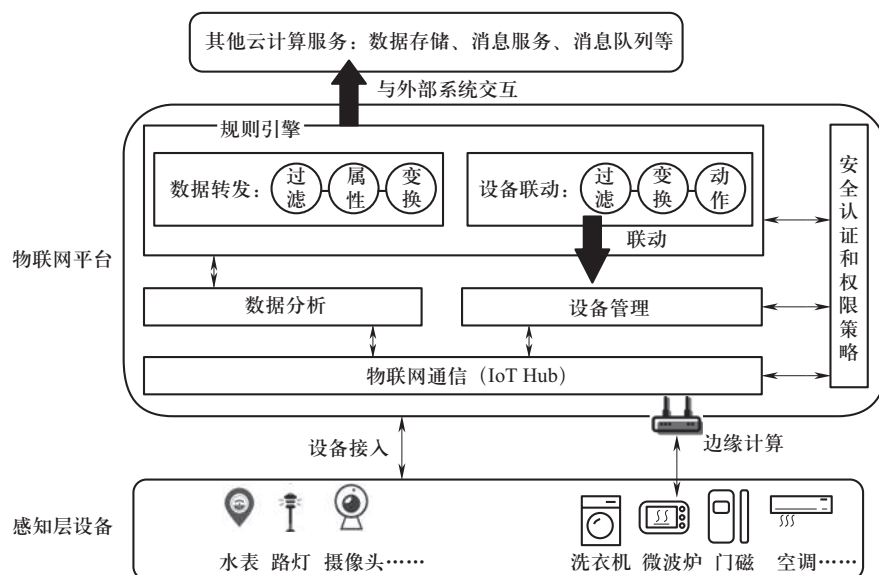
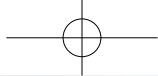


图 2-1 规则引擎功能图

规则链就是规则及其关联关系的集合。在物联网平台的所有模块中，规则链是最复杂和最灵活的模块。规则链的应用设计也是应用开发方向的物联网工程技术人员必



会的技能之一。本章以告警规则链的设计过程为例讲解规则节点的使用以及如何使用规则链进行数据处理和转发。

- **职业功能：** 物联网平台应用开发。
- **工作内容：** 规则链应用设计。
- **专业能力要求：** 能使用规则节点对接入的传感数据进行处理；能根据规则链设计文档，实现规则链中的数据转发。
- **相关知识要求：** 传感数据结构知识、规则链设计知识。



第一节 规则链应用基础

规则引擎是指用户在物联网平台上对接入平台的设备设定相应的规则，在条件满足所设定的规则后，平台会触发相应的动作来满足用户需求。本节将在 ThingsBoard 上建立一个实时监控粉尘浓度的规则链，该规则链处理 HTTP 协议传入的消息，并在规则链中对传入的传感数据进行处理。读者通过以上处理过程来学习规则链的基本使用方法。

考核知识点及能力要求：

- 了解规则引擎；
- 能查阅文档了解规则节点的分类；
- 了解规则节点间的关联关系；
- 能创建、编辑、链接和调试规则链；
- 能添加、修改、删除规则节点；
- 能接入 HTTP 协议的设备数据；
- 能解决规则链使用过程中出现的问题。

一、规则引擎概述

ThingBoard 内置的规则引擎可以接收设备的消息，还可以通过自定义的规则实现处理和转发，其体系结构如图 2-2 所示。

规则引擎中主要包含消息、规则节点和规则链。在学习具体的规则链设计前，需要先了解这些基本概念。

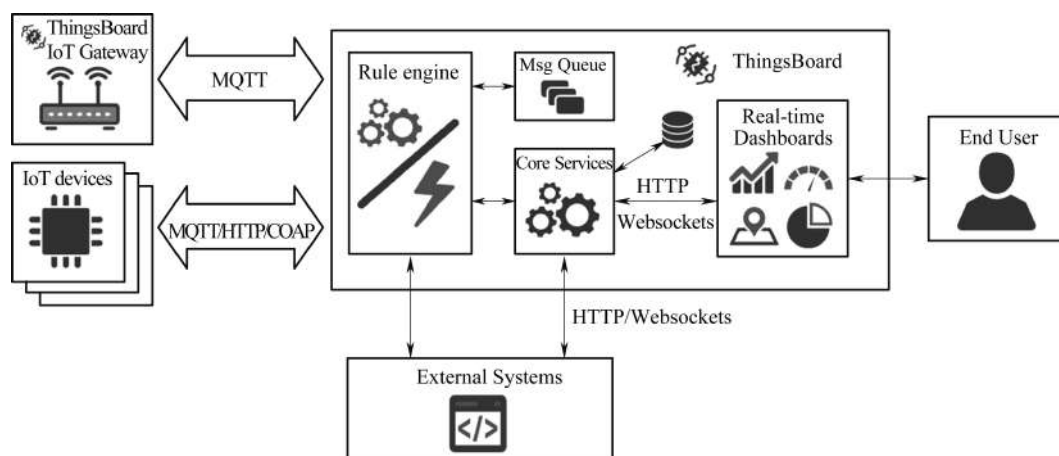


图 2-2 ThingsBoard 的规则引擎体系结构

（一）规则引擎消息

消息用于表示系统中的各种事件，有来自设备的传入遥测（实时上报的数据）、通过 RESTful 请求规则引擎事件、远程过程调用（Remote Procedure Call，RPC）请求事件、告警事件、实体生命周期事件（创建、更新、删除、分配、取消分配、属性更新、属性删除）设备状态事件（连接、断开、活动、非活动）等。

消息是可以被序列化的、不可变的数据结构。消息包含的信息如下：

消息 ID：基于时间的通用唯一标识符。

消息的发起者：设备、资产或其他实体标识符。

消息类型：发布遥测或告警类型等。

消息的有效载荷：带有实际消息有效载荷。

元数据：包含有关消息的附加数据的键值对列表。

（二）规则节点

规则节点一次处理一个传入消息，并生成一个或多个传出消息。ThingsBoard 中的规则节点类型有过滤、属性（集）、变换、动作、外部的、规则链等，如图 2-3 所示。

（三）规则节点关联

规则节点之间可以关联，每种关联都有关联类型。关联类型表示该关联的逻辑意思的名称。规则节点在生成输出消息时，通过指定关联类型将生成的消息路由到下一个节点。

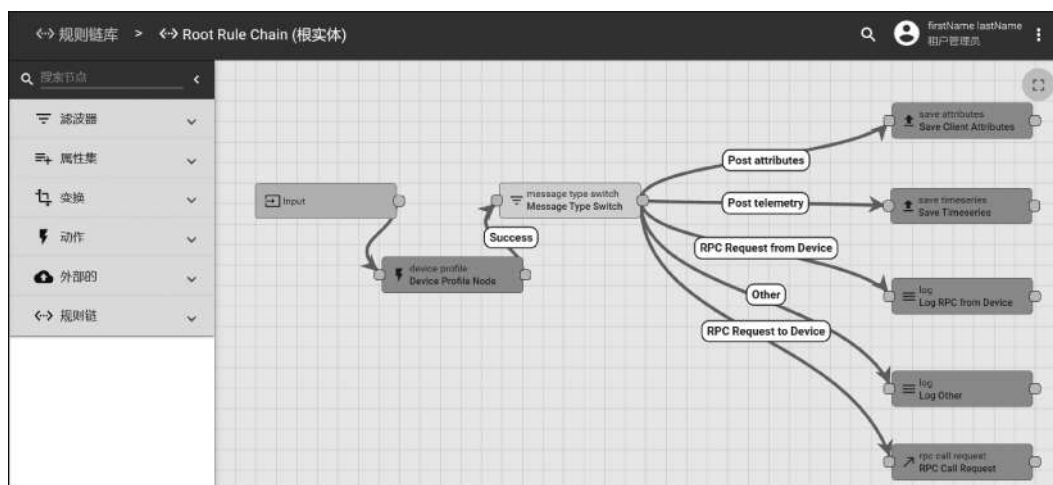


图 2-3 ThingsBoard 中的规则节点类型

规则节点关联类型可以是“Success”“Failure”等，也可以是“True”“False”“Post Telemetry”“Attributes Updated”“Entity Created”等。

（四）规则链

规则链是规则节点及其关系的集合。将针对特定数据包的规则节点，按照顺序依次放入对应的链中，节点之间通过线来互相连接，于是，来自规则节点的出站消息将依次发送到下一个规则节点中。

租户管理员可以定义一个根规则链，还可以定义多个其他规则链。根规则链处理所有传入的消息，并将其转发到其他规则链以进行其他处理。其他规则链也可以将消息转发到不同的规则链。

ThingsBoard 的规则链可以执行的操作有：在保存到数据库之前，对传入遥测（遥测也就是常说的设备测量状态，如温度计的温度、灯的亮度等）或属性进行数据验证和修改；将遥测或属性从设备复制到相关资产，以便汇总遥测；根据定义的条件创建、更新、清除警报；根据设备生命周期事件触发操作；加载处理所需的其它数据；触发对外部系统的 RESTful 调用；发生复杂事件时发送电子邮件；根据定义的条件进行 RPC 调用；与外部系统（如 Kafka、RabbitMQ 等）集成；等等。



二、规则链基本使用

规则链的基本使用包含创建、编辑、链接和调试。接下来以创建实时监控粉尘浓度的规则链为例，学习规则链的基本用法。

（一）创建和编辑规则链

1. 根规则链

在规则链库中，每个租户都有一条名为“Root Rule Chain”的根规则链，用来处理该租户下的设备数据，如图 2-4 所示。

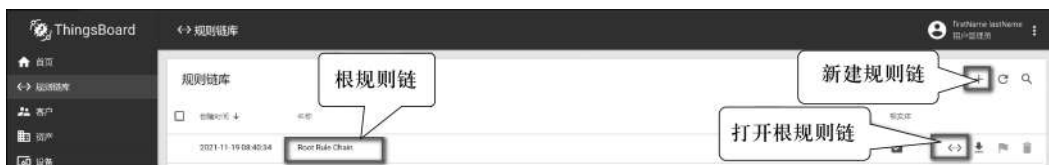


图 2-4 ThingsBoard 上的根规则链

打开根规则链，可以看到默认所有传入数据都会经过根规则链进行处理，如图 2-5 所示。

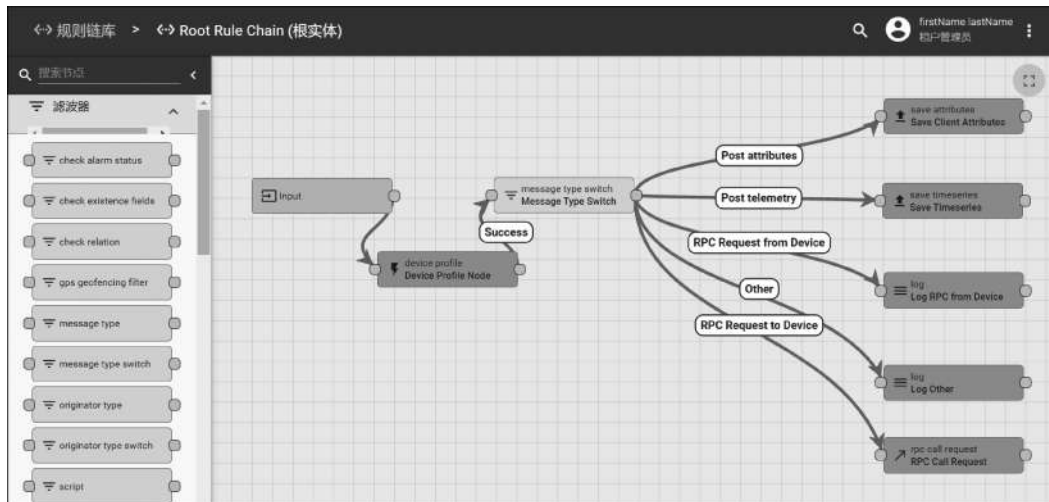


图 2-5 通过根规则链传入的数据

2. 创建并打开新规则链

单击图 2-4 右上角的“+”，在弹出的页面中选择“创建新的规则链”，在“添加

规则”对话框输入规则名称“测试 PM25”，勾选调试模式，描述部分的内容可选填，单击“添加”按钮，即可创建新的规则链，如图 2-6 所示。



图 2-6 新建规则链

创建好规则链后，返回规则链库，查看创建好的新规则链“测试 PM25”是否存在，如图 2-7 所示。



图 2-7 查看新规则链“测试 PM25”

如果想打开创建好的规则链，只要单击要打开的规则链，在右侧弹出的页面中单击“Open Rule Chain”，就可以打开规则链。

3. 规则链链接

规则链之间通过彼此链接，可以将一条链的处理结果传输到另一条链继续处理。打开根规则链“Root Rule Chain”，在左侧节点列表中找到节点规则链“rule chain”，拖拽到右侧面板，选择规则链“测试 PM25”，单击“添加”按钮，连接时序节点与规则链节点，选择链接标签“Success”，则表示时序数据保存成功后，进入规则链“测试 PM25”，如图 2-8 所示。

4. 规则链调试

打开规则链的调试模式用于调试规则。调试完成后，记得关闭规则链的调试模式，否则会增加数据库存储压力。

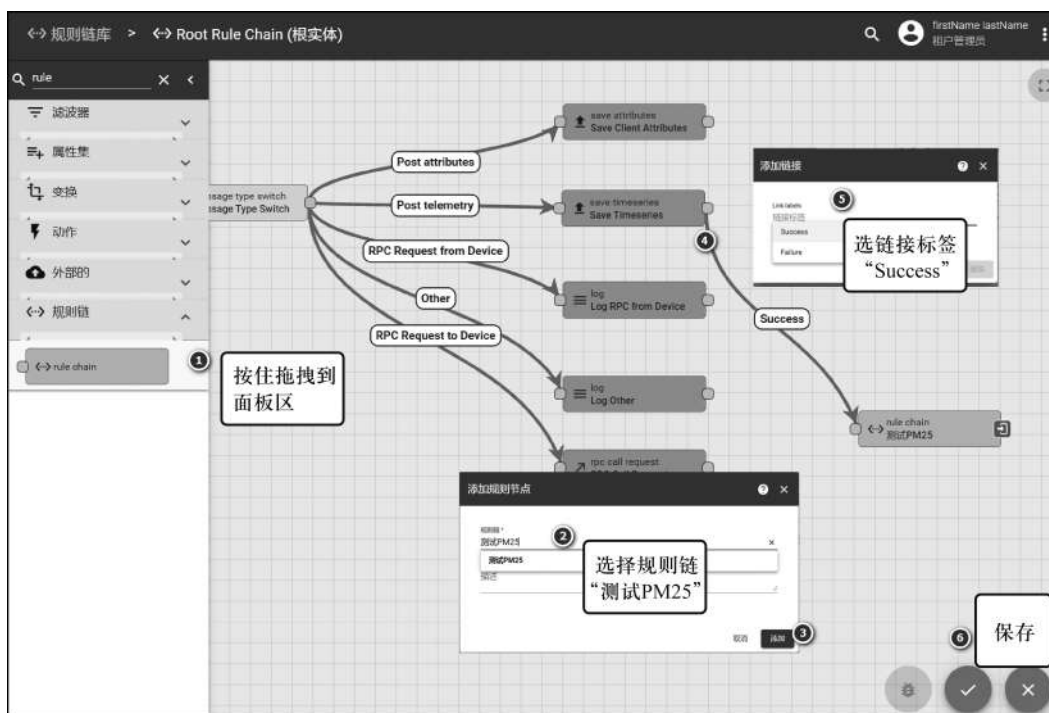


图 2-8 链接根规则链与“测试 PM25”规则链

（二）编辑规则节点

这里以在规则链“测试 PM25”中添加一个名为“消息变换”的规则节点，实现不管传入的粉尘浓度是什么数值，经过“消息变换”规则节点后，该值都会被设置 8 后再传到下一个规则节点为例，讲解规则节点的创建过程。

假设希望粉尘浓度的消息中“key 为 pm25”的传感数据经过规则节点处理后，传感数据值都变换为 8，请遵循以下操作步骤。

第一步：打开规则链“测试 PM25”，在左侧规则节点列表中拖拽节点“script”（注意是变换列表下的“script”）到编辑区。在弹出页中填写名称“消息变换”。

第二步：在“消息变换”规则节点的代码区增加一行代码“msg.pm25=8;”，这行代码的意思是修改消息中“key 为 pm25”的值为 8，则所有的“key 为 pm25”的传感数据经过该规则节点后，传感值都会被修改为 8，然后再传给下一个规则节点。

第三步：单击“添加”按钮，即可生成规则节点。

第四步：将“消息变换”规则节点与“Input”规则节点相连后，保存规则链，即可完成规则链的创建，如图 2-9 所示。

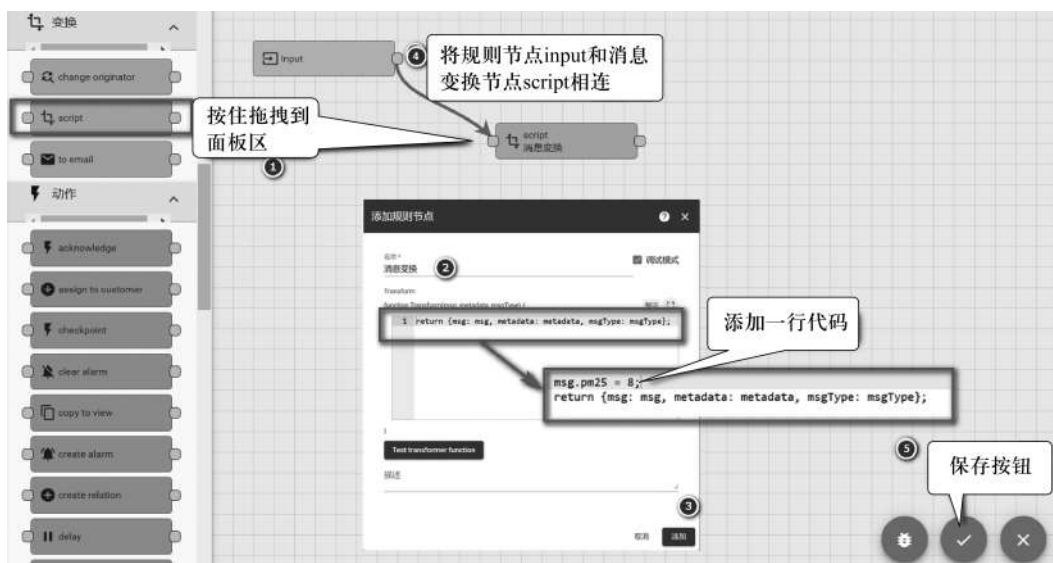


图 2-9 创建规则链

如果需要修改或删除规则节点，可以双击规则节点，按提示进行操作。需要注意的是，所有修改都需要保存才能生效。

（三）HTTP 协议设备接入

设备接入并不一定需要真正的设备，可以通过程序或者命令来模拟设备，然后通过模拟设备命令来更新设备的状态（也称遥测值），从而实现向物联网平台发送信息。

在 ThingsBoard 上添加设备实体，需要指定设备的设备配置（Device profiles）。设备配置是设备类型或者是设备类型的配置，定义了设备的规则链、队列、传输配置、告警规则等参数。创建设备时，如果不指定设备配置，将默认使用“Default”设备配置。

1. 添加设备配置

添加设备配置要先选中左侧菜单栏的“Device profiles”，然后单击右上角的“+”，在弹出页中输入设备配置名称为“my_pm25”，规则链选择上面创建好的“测



试 PM25”，队列选择“Main”，单击“添加”按钮即成功添加设备配置，如图 2-10 所示。



图 2-10 添加设备配置

创建好的设备配置文件可以到“Device profiles”中进行查看。

2. 创建传感设备

按下述操作步骤创建传感设备“pm25”，设备配置文件选上一步创建好的“my_pm25”，如图 2-11 所示。



图 2-11 创建“pm25”传感设备

查看添加好的设备信息，如图 2-12 所示。



图 2-12 查看设备信息

3. 通过 HTTP 协议发送遥测传感数据

通过 HTTP 协议发送设备的遥测数据，请遵循以下操作步骤。

第一步：获取“pm25”设备的访问令牌。单击“pm25”设备，在设备详细信息页，复制访问令牌以便后续使用，如图 2-13 所示。



图 2-13 获取“pm25”设备的访问令牌

第二步：使用命令行工具 curl 通过 HTTP 协议发送传感数据。ThingsBoard 为 IoT 设备提供了 HTTP、MQTT、CoAP 三种方式发送传感数据。以 HTTP 方式为例，IoT 设备发送 HTTP 协议的遥测数据需要的参数见表 2-1。

表 2-1 以 HTTP 方式发送遥测数据的参数	
参数	值
url	http: //\$HOST/api/v1/\$ACCESS_TOKEN/telemetry
method	POST
header	Content-Type: application/json
json	{ “键”: 值 }



其中，\$HOST 是 ThingsBoard 的 URL，\$ACCESS_TOKEN 是设备的访问令牌，HOST 和 PORT 的值以实际使用时分配的为准。

使用在 Linux/Mac/Windows10 平台上都有的命令行工具 curl，发送 HTTP 协议的传感数据的命令为：

```
curl -i -X POST --header "Content-Type: application/json" -d '{"键": 值}'  
http://$HOST/api/v1/$ACCESS_TOKEN/telemetry
```

假设设备的访问令牌为 2c8H0gMtE0F4IRj8PYIr，HOST 为 tb.nlecloud.com，想发送粉尘浓度的值为 24，则具体的 curl 的命令为：

```
curl -i -X POST -d '{"pm25": "24"}'  
http://tb.nlecloud.com/api/v1/2c8H0gMtE0F4IRj8PYIr/telemetry --  
header "Content-Type: application/json"
```

需要注意的是，在 win10 系统发送上述命令时，单引号要改成双引号。打开 CMD 命令窗口，分别发送 pm25 的值为 24 和 48 的遥测数据，如图 2-14 所示。

```
C:\Users\pkr>curl -i -X POST -d '{"pm25": "24"}' http://tb.nlecloud.com/api/v1/2c8H0gMtE0F4IRj8PYIr/telemetry  
-header "Content-Type: application/json"  
HTTP/1.1 200  
Content-Length: 0  
Date: Fri, 24 Dec 2021 06:59:31 GMT  
  
C:\Users\pkr>curl -i -X POST -d '{"pm25": "48"}' http://tb.nlecloud.com/api/v1/2c8H0gMtE0F4IRj8PYIr/telemetry  
-header "Content-Type: application/json"  
HTTP/1.1 200  
Content-Length: 0  
Date: Fri, 24 Dec 2021 06:59:44 GMT
```

图 2-14 使用 HTTP 协议发送遥测数据

（四）查看经过规则链处理后的数据

打开规则链“测试 PM25”，双击规则节点“消息变换”，可以看到消息类型有“IN”和“OUT”，分别代表传入的消息和经过变换后的输出消息。传入数据为 24 和 48，经过变换后的值都变为 8，说明变换规则节点对数据进行了过滤和处理，如图 2-15 所示。



图 2-15 查看规则链处理后的数据

第二节 规则链设计

本节先介绍各种类型规则节点的使用方法，再以告警规则链的设计过程为例，介绍如何使用不同的规则节点处理传入的消息，以及如何将传入的消息转发到指定规则链。

考核知识点及能力要求：

- 了解不同类型的规则节点的功能；
- 能查阅文档了解各种规则节点如何处理传入的消息；
- 能使用合适的规则节点创建告警规则链；
- 能将处理后的消息转发到指定规则链；



- 能解决规则链使用过程中出现的问题。

一、使用规则节点处理传入消息

规则链中的规则节点有多种类型，不同的规则节点处理消息的用法不同。本节将学习如何使用不同的规则节点对传入消息进行处理。为方便描述，添加各个规则节点时，名称均按节点名字确定，实际应用开发时需改成有真实意义的规则节点名称。同时，因篇幅有限，仅列举常用的规则节点的用法，学习过程中，鼠标悬停在每个规则节点上时，会出现该规则节点的具体用法提示，需要了解更多规则节点用法的读者可以到官网查阅文档。

（一）过滤节点

过滤节点用于消息过滤和路由，使用配置条件筛选过滤传入的消息，过滤成功走真链，过滤错误走假链。

1. 消息类型过滤节点

系统中有预定义的消息类型，如“Post attributes”“Post telemetry”“RPC Request”等。在消息类型过滤节点的配置中，管理员为传入的消息定义了一组允许的消息类型，如果是允许传入的消息类型，则消息通过 True 链发送，否则通过 False 链发送，如图 2-16 所示。

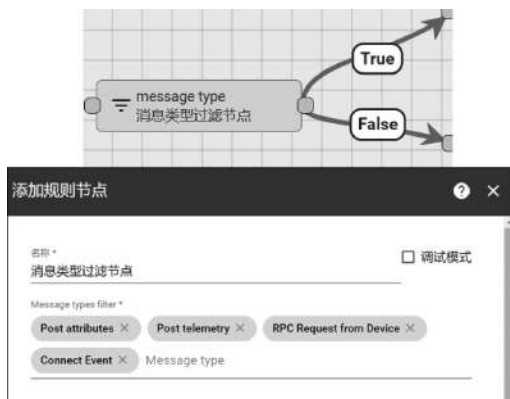


图 2-16 消息类型过滤节点

2. 消息类型切换节点

消息类型切换节点用于按消息类型路由传入消息，如果传入的消息是已知消息类

型，则将其发送到相应的链，否则将其消息发送到其他链。如果使用自定义消息类型，那么可以通过其他消息类型切换节点将这些消息发送到配置了所需路由逻辑的切换节点或消息类型过滤器节点，如图 2-17 所示。

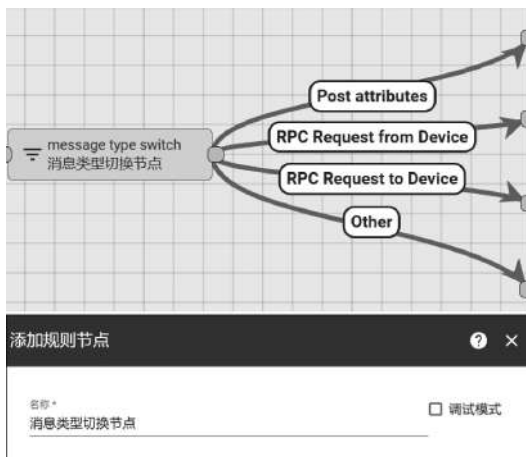


图 2-17 消息类型切换节点

3. 脚本过滤节点

脚本过滤节点使用 JavaScript 函数评估传入的消息。JavaScript 函数接收三个输入参数，说明如下：

msg: 消息有效负载。消息有效负载可以通过 msg 变量访问，例如 msg.temperature > 20。

metadata: 消息元数据。通过 metadata 变量访问消息元数据，例如 metadata.deviceType === 'vehicle'。

msgType: 消息类型。消息类型可以通过 msgType 变量访问，例如 msgType === 'POST_TELEMETRY_REQUEST'。

JavaScript 函数返回布尔值，如果为真，则通过 True 链发送消息，否则使用 False 链发送消息，如图 2-18 所示。双击规则节点，编辑完 JavaScript 函数后，可以单击“Test filter function”进行验证，如图 2-19



图 2-18 脚本过滤节点



所示。在“测试脚本功能”页面输入参数并验证 JavaScript 函数的输出，如图 2-20 所示。

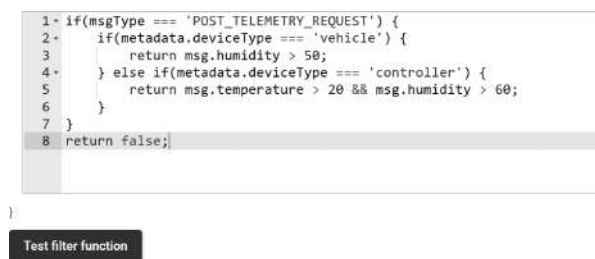


图 2-19 验证 JavaScript 函数

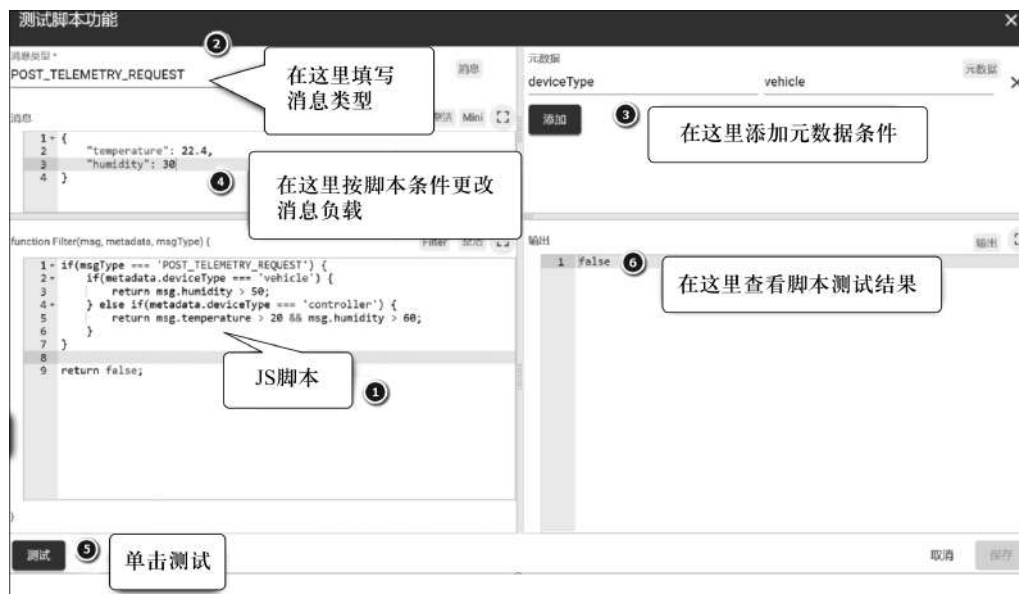


图 2-20 测试脚本

4. 切换节点

切换节点用于将传入消息路由到一个或多个输出链，节点执行配置好的 JavaScript 函数，该函数接收的输入参数与脚本过滤节点相同。该脚本应该返回一个下一个关系名称的数组，其中含有路由消息。如果返回数组为空，则消息将不会被路由到任何节点并被丢弃，如图 2-21 所示。

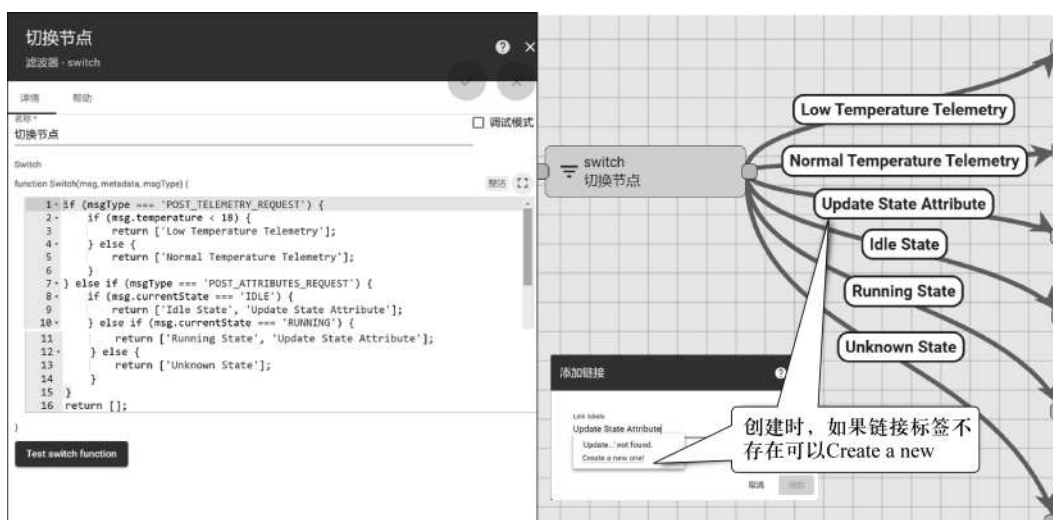


图 2-21 切换节点

(二) 属性节点

属性节点用于更新传入消息的元数据（Metadata），常用的有客户属性、发起人字段等。

1. 客户属性

客户属性节点找到消息发起者实体的客户并将客户属性或最新遥测值添加到消息元数据中。管理员可以配置原始属性名称和元数据属性名称之间的映射关系。如果选中“Latest telemetry”复选框，节点将获取最新遥测数据，否则节点将获取服务器范围属性，出站消息元数据将包含配置的属性；客户属性允许消息发起者的类型有“Customer”“User”“Asset”“Device”，如果发现不受支持的发起者类型，则会引发错误；如果发起者没有分配客户实体，则使用 Failure 链，否则使用 Success 链，如图 2-22 所示。

2. 发起人字段

节点获取消息发起者实体的字段值并将它们添加到消息元数据中，管理员可以配置字段名称和元数据属性名称之间的映射关系。如果指定的字段不是 Message Originator 实体字段的一部分，将被忽略。允许的消息发起者类型有“租户”“客户”“用户”“资产”“设备”“警报”“规则链”。如果找到不受支持的发起者类



型，则使用 Failure 链，否则使用 Success 链；如果未找到字段值，则不会将其添加到消息元数据中，但仍会通过 Success 链进行路由。出站消息元数据将仅包含配置的属性，如图 2-23 所示。



图 2-22 客户属性

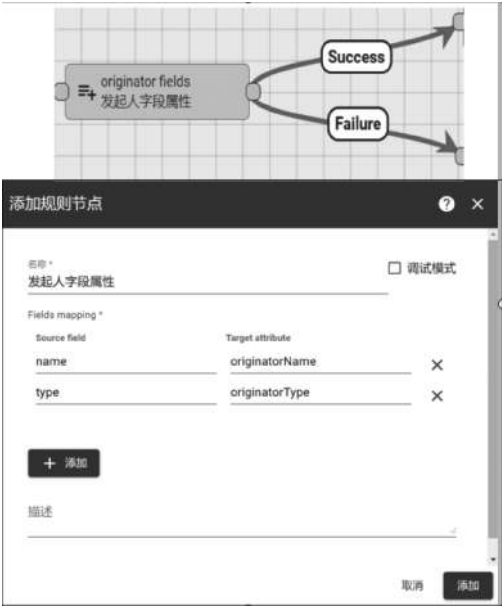


图 2-23 发起人字段

（三）变换节点

变换节点用于更改传入消息字段，如发起者、消息类型、有效负载和元数据。变换节点包括脚本变换节点、到电子邮件节点等。

1. 脚本变换节点

脚本变换节点使用配置的 JavaScript 函数更改有效负载、元数据或消息类型。JavaScript 函数接收 msg、metadata、msgType 三个参数，处理完后，JavaScript 函数应返回的结果对象结构如下：

```
{ return msg: new payload, metadata: new metadata, msgType: new msgType }
```

结果对象中的所有字段都是可选的，如果未指定，将从原始消息中获取。来自该节点的出站消息将是使用配置的 JavaScript 函数构造的新消息。

假设某个节点的原始消息类型为“POST_TELEMETRY_REQUEST”，其原始元数据属性为 { "sensorType": "temperature" }，如果想要将消息类型更改为“CUSTOM_UPDATE”，将附加属性版本添加到值为 v1.1 的有效负载中，将元数据中的 sensorType 属性值更改为 roomTemp，则使用脚本转换节点的 JavaScript 函数代码，如图 2-24 所示。



图 2-24 修改脚本转换节点

2. 到电子邮件节点

通过使用从消息元数据派生的值填充电子邮件字段，将消息转换为电子邮件消息。所有电子邮件字段都可以配置为使用元数据中的值。

(四) 动作节点

动作节点用于根据传入的消息执行各种动作。

1. 创建报警节点

创建报警节点尝试加载具有为消息发起者配置的警报类型的最新警报。如果存在未清除警报，则该警报将被更新，否则将创建一个新警报，如图 2-25 所示。

2. 清除报警节点

清除报警节点加载具有为消息发起者配置的警报类型的最新警报，如果存在则清除警报，如图 2-26 所示。

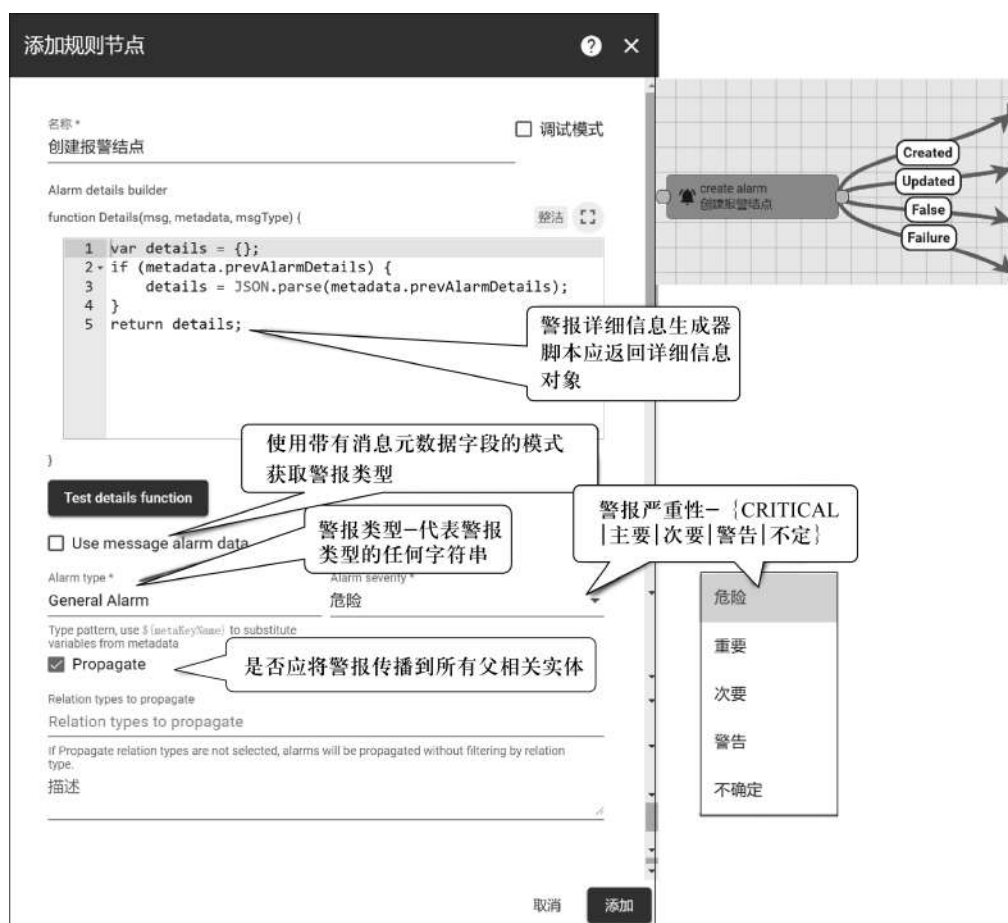


图 2-25 创建报警节点

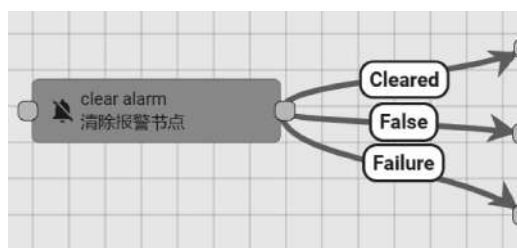


图 2-26 清除报警节点

此节点更新当前警报时，如果已经确认，则将警报状态更改为 CLEARED_ACK，否则更改为 CLEARED_UNACK，同时将清除时间设置为当前系统时间。如果告警不存在或已清除告警，则原始 Message 将通过 False 链传递给下一个节点，否则新消息将通过 Cleared 链传递。

(五) 外部节点

外部节点将消息及数据路由到外部中间件，或者其他第三方云平台中，用于与外部系统进行交互，如图 2-27 所示。



图 2-27 与外部进行交互的节点

二、设计告警规则链

本节学习如何使用过滤、创建告警和清除告警规则节点对传入数据进行处理。

假设上面创建的传感器设备收集粉尘读数并将其推送到 ThingsBoard，粉尘轻度污染的标准为 100 ~ 150，重度污染的标准为 200 ~ 300，规则链要求实现如果 msg.pm25 ≥ 200，则创建或更新现有告警；如果 msg.pm25 < 200，则清除告警。

(一) 创建告警规则链并指定设备配置

切换到规则链库，创建新的规则链，名称为“PM25 告警链”，并开启调试模式。修改设备配置“my_pm25”的规则链为“PM25 告警链”，如图 2-28 所示。

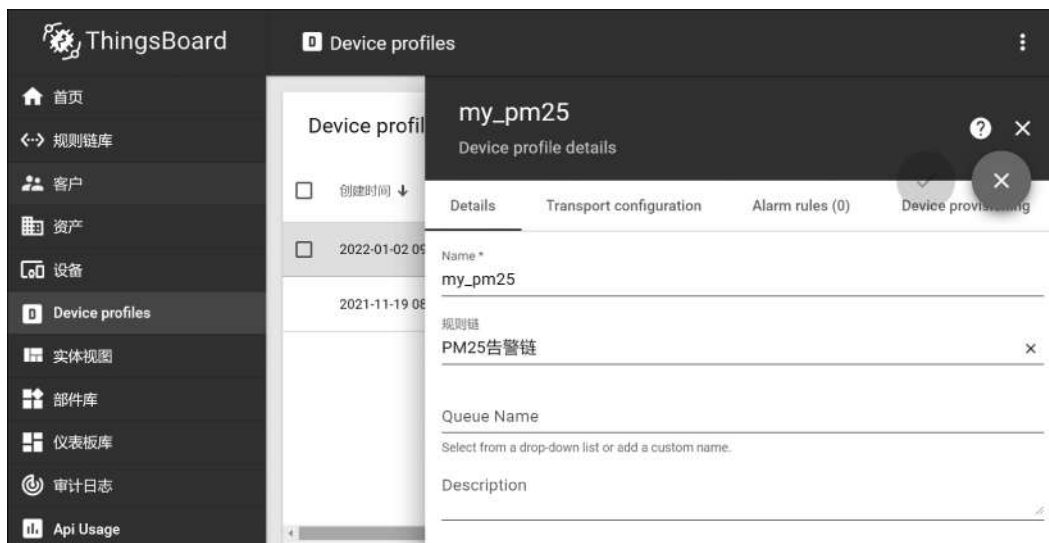


图 2-28 修改设备配置“my_pm25”的规则链

（二）使用规则节点处理传入数据

打开规则链“PM25 告警链”，添加 3 个节点，如图 2-29 所示。

针对要添加的 3 个节点的说明如下：

（1）节点 A 为过滤脚本节点，如果 msg.pm25 的值在预期的间隔内，脚本将返回 false，否则将返回 true。

（2）节点 B 为创建告警节点，如果 msg.pm25 的值不在预期的时间范围内，则创建或更新告警，过滤器脚本节点返回 true。

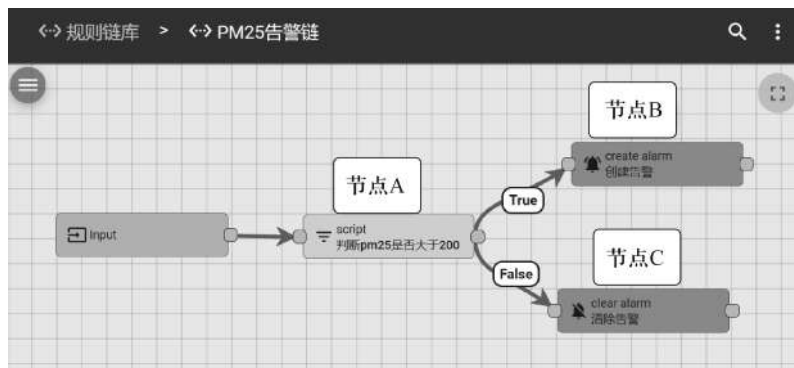


图 2-29 打开规则链“PM25 告警链”中的节点

(3) 节点 C 为清除告警节点，如果 msg.pm25 的值在预期的时间范围内，脚本节点返回 false，否则清除告警。

1. 添加节点 A（脚本过滤）

节点 A 是脚本过滤节点，添加 “script” 节点并将其连接到 “Input” 节点。该节点将使用以下脚本验证：

```
return msg.pm25 > 200;
```

如果 pm25 的值超过 200，脚本将返回 true，否则将返回 false，如图 2-30 所示。



图 2-30 添加过滤节点

单击 “Test filter function”，测试脚本函数是否正确，分别输入 300 和 150，测试脚本是否能正确返回对应的值，操作如图 2-31、图 2-32 所示。

2. 添加节点 B（创建告警）

添加 “Create Alarm” 节点并将其连接到关系类型为 “True” 的节点 A。如果 msg.pm25 的值大于 200（节点 A 返回 true），将产生告警或更新告警。填写名称为 “创建告警”，警报类型填写 “PM25 Alarm”，勾选 “Propagate”，以将告警传播给相关实体，开启调试模式，如图 2-33 所示。

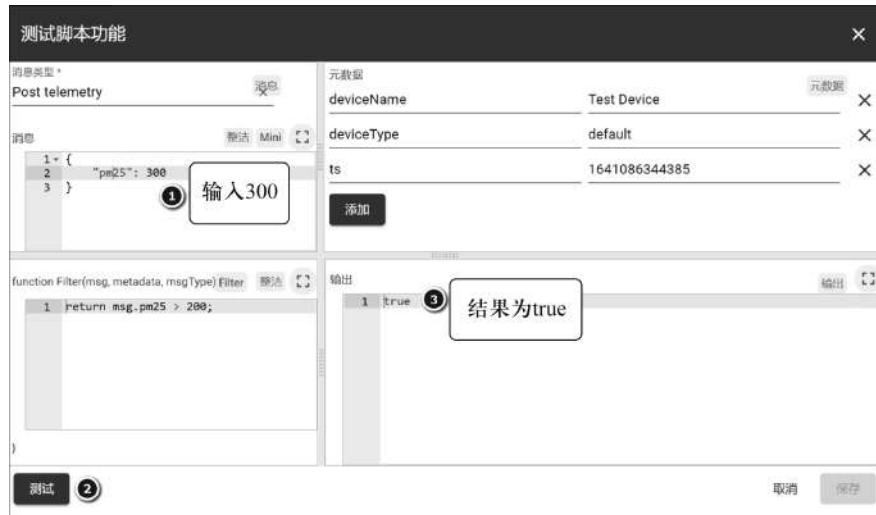


图 2-31 测试脚本返回 true



图 2-32 测试脚本返回 false

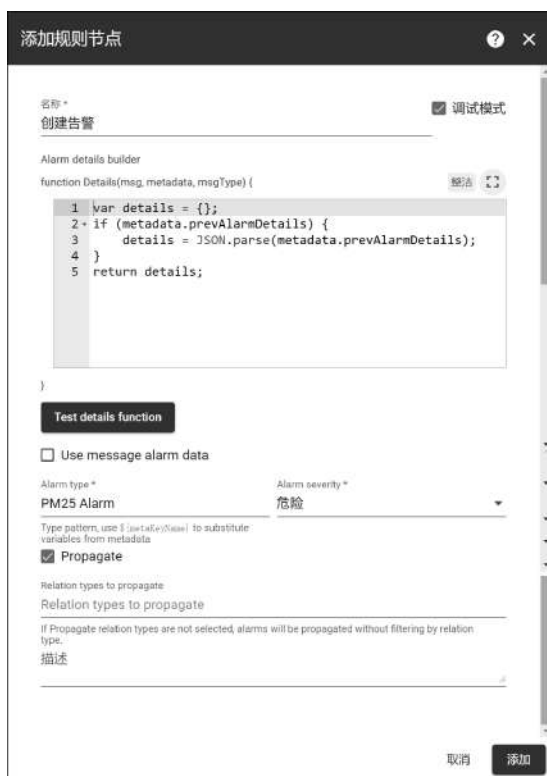


图 2-33 创建告警节点

3. 创建节点 C（清除告警）

添加“Clear Alarm”节点并将其连接到关系类型为“False”的节点 A。如果 msg.pm25 的值小于 200（节点 A 返回 false），将清除告警。填写名为“清除告警”，告警类型写为“PM25 Alarm”，开启调试模式，如图 2-34 所示。

（三）将传入消息转发到指定的规则链

在根规则链中添加节点 D（Rule Chain 节点），以将传入并经过处理的遥测数据转发到“PM25 告警链”规则链，如图 2-35 所示。

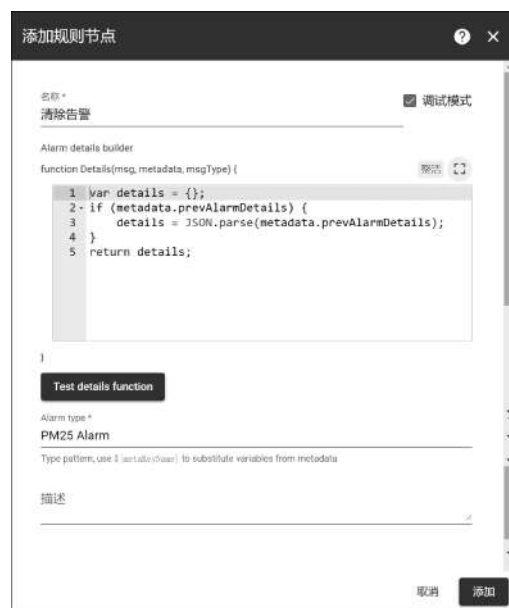


图 2-34 添加清除告警节点

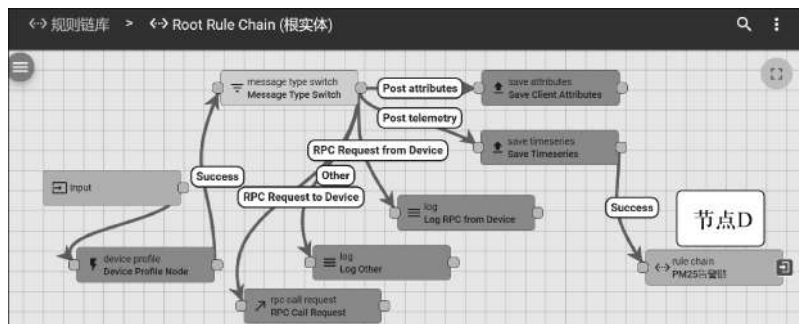


图 2-35 将传入消息转发到“PM25 告警链”

（四）验证规则链

1. 验证告警和更新

发送 HTTP 协议数据，数据值为 333，让规则链产生告警，命令如下：

```
// 发送产生告警的数据
curl -i -X POST -d '{"pm25": 333}'
http://tb.nlecloud.com/api/v1/2c8H0gMtE0F4IRj8PYlr/telemetry --header
"Content-Type: application/json"
```

发送成功后，打开设备的属性界面，服务端属性“active”的值是“true”，说明设备连接成功，如图 2-36 所示。再次发送同样的告警数据，会产生告警更新。打开“创建告警”节点，观察关系类型为“Created”和“Update”的事件，如图 2-37 所示。



图 2-36 验证告警的产生信息

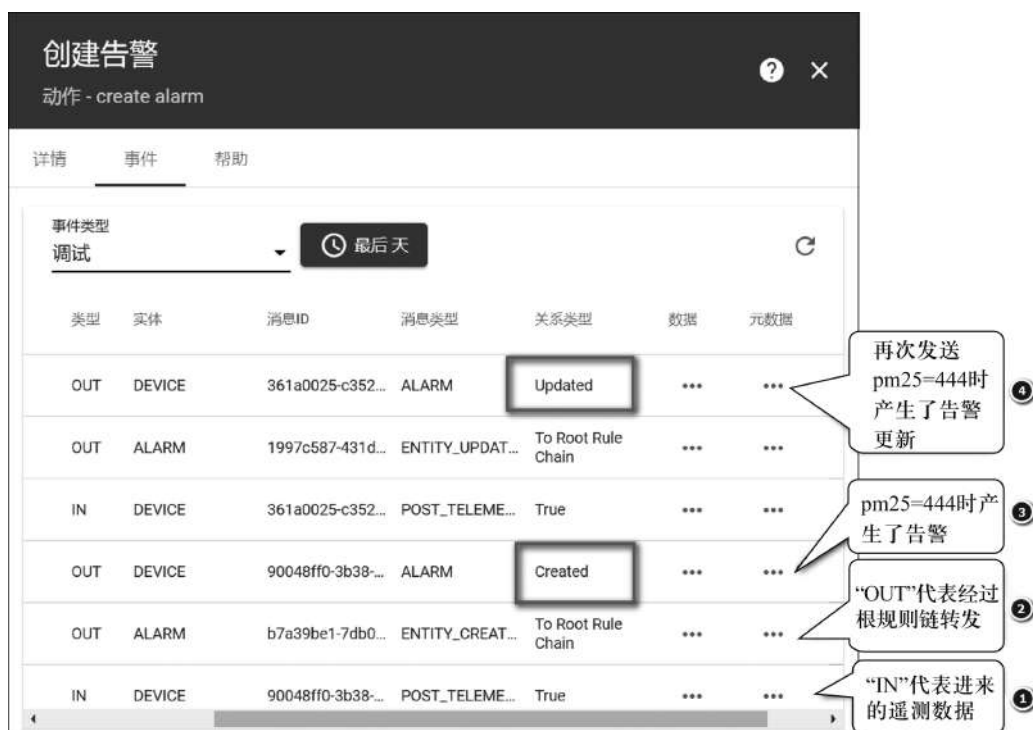


图 2-37 验证告警的更新信息

2. 验证告警清除

发送 HTTP 协议数据，数据值为 188，让告警链产生“告警清除”的事件，命令如下：

```
curl -i -X POST -d '{"pm25": 188}'  
http://tb.nlecloud.com/api/v1/2c8H0gMtE0F4IRj8PYlr/telemetry --header  
"Content-Type: application/json"
```

打开“清除告警”节点，观察关系类型为“Cleared”的事件，如图 2-38 所示。
再次打开设备“pm25”，观察“警告”项，如图 2-39 所示。

到目前为止，本章已经实现了规则链的基本使用和规则链的设计方法，并设计了告警规则链，更多的规则链用法请读者查阅官方文档。



图 2-38 验证告警的清除信息



图 2-39 查看告警信息

思考题

1. 规则引擎包含哪些部分？
2. 如何发送 HTTP 协议数据？
3. 规则节点的用途是什么？
4. 如何使用脚本过滤节点？
5. 如何把规则链链接到根规则链？
6. 简述告警规则链的创建、更新和清除的流程。