

用 ESP32 访问阿里云

这是一个使用 esp32 运行 microPython 通过 MQTT 上报温度与湿度数据到阿里云服务器的教程。

阿里云作为一个云服务器, 支持下位机通过 mqtt 协议进行访问与数据通信。

先放上教程来源：

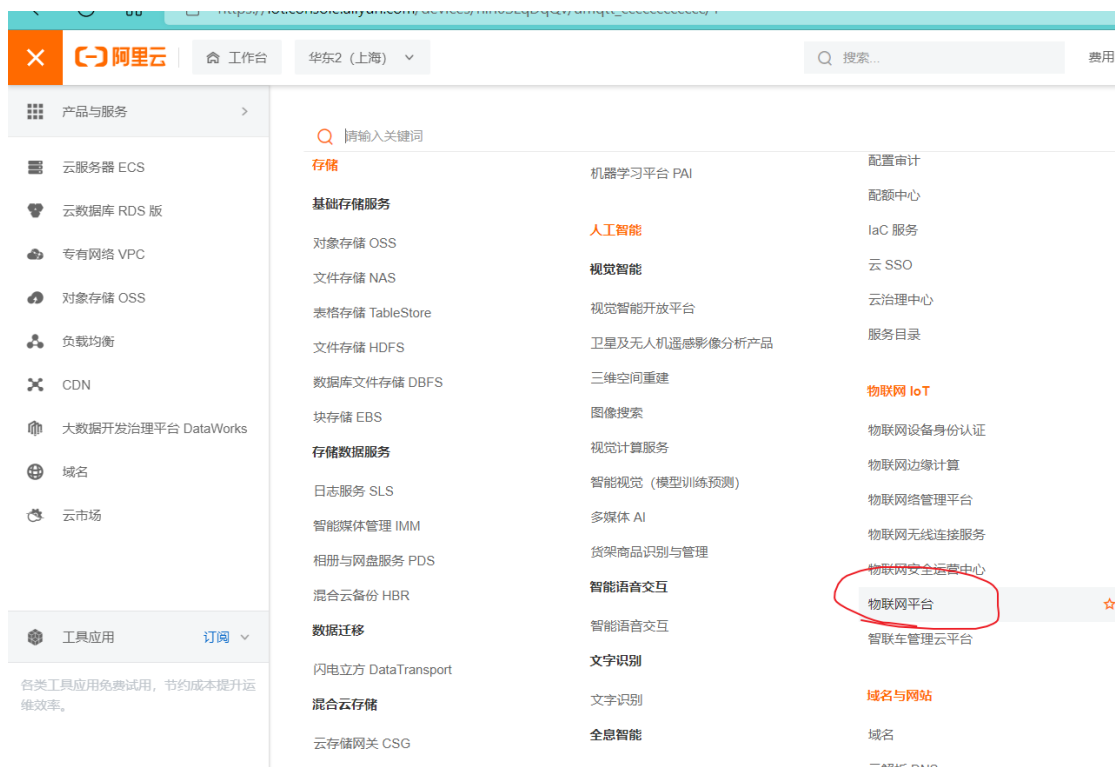
https://blog.csdn.net/freemote/article/details/105322722?utm_source=app&app_version=5.3.1&code=app_1562916241&uLinkId=usr1mkqgl919blen

细节部分可以参考这一篇 CSDN 博客。这里只介绍大部分必要的步骤。

以及王铭东老师的 esp32 的教程文档, 这里面对于 mqtt 通讯协议的介绍非常详细, 大家可以去看一看

<http://doc.3dcarve.com/0006.zhishi.esp32/02.doc/index.html#/c06.mqtt2>

大家需要先自行登录阿里云官网 找到产品与服务的物联网平台



大家需要提前注册一个阿里云账号

先添加一个产品

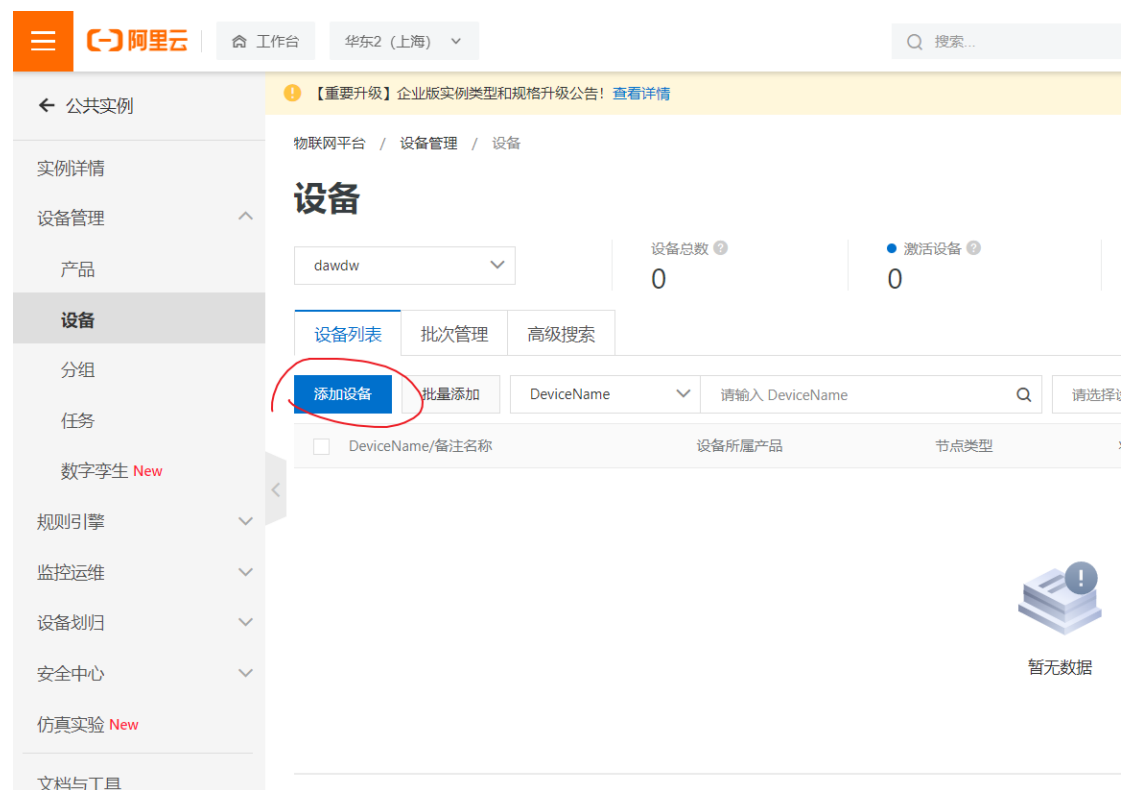
The screenshot shows the 'Add Product' configuration page in an IoT platform. The left sidebar contains navigation options: 概览, 设备管理, 产品, 设备, 分组, CA 证书, 规则引擎, 监控运维, 边缘计算, 视频服务, IoT Studio, 数据分析, 应用托管, 实例管理, and 产品文档. The '产品' (Product) section is active. The main configuration area includes the following fields and options, all of which are circled in red:

- 产品名称** (Product Name): A text input field containing 'my-mqtt-device'.
- 所属品类** (Product Category): Radio buttons for ☐ 标准品类 (Standard Category) and ☒ 自定义品类 (Custom Category).
- 节点类型** (Node Type): Three selectable icons: **直连设备** (Direct Device), **网关子设备** (Gateway Sub-device), and **网关设备** (Gateway Device). The '直连设备' option is selected.
- 连网方式** (Network Method): A dropdown menu showing 'WiFi'.
- 数据格式** (Data Format): A dropdown menu showing 'ICA 标准数据格式 (Alink JSON)'.
- 认证方式** (Authentication Method): A section header with a dropdown arrow.
- 更多信息** (More Information): A section header with a dropdown arrow.
- 产品描述** (Product Description): A section header with a dropdown arrow.
- 保存** (Save) and **取消** (Cancel) buttons at the bottom.

<https://blog.csdn.net/freemote>

按照 csdn 博客上面说的来就 ok

添加产品过后 进入添加设备的界面



对于产品模型的一些介绍

物模型指将物理空间中的实体数字化，并在云端构建该实体的数据模型。在物联网平台中，定义物模型即定义产品功能。完成功能定义后，系统将自动生成该产品的物模型。物模型描述产品是什么，能做什么，可以对外提供哪些服务。

物模型，简称 TSL，即 Thing Specification Language。是一个 JSON 格式的文件。它是物理空间中的实体，如传感器、车载装置、楼宇、工厂等在云端的数字化表示，从属性、服务和事件三个维度，分别描述了该实体是什么，能做什么，可以对外提供哪些信息。定义了这三个维度，即完成了产品功能的定义。

物模型将产品功能类型分为三类：属性、服务和事件。定义了这三类功能，即完成了物模型的定义。

功能类型

属性 (Property)

一般用于描述设备运行时的状态，如环境监测设备所读取的当前环境温度等。属性支持 GET 和 SET 请求方式。应用系统可发起对属性的读取和设置请求。

服务 (Service)

设备可被外部调用的能力或方法，可设置输入参数和输出参数。相比于属性，服务可通过一条指令实现更复杂的业务逻辑，如执行某项特定的任务。

事件 (Event)

设备运行时的事件。事件一般包含需要被外部感知和处理的通知信息，可包含多个输出参数。如，某项任务完成的信息，或者设备发生故障或告警时的温度等，事件可以被订阅和推送。

上面这三部分是在初始化自定义功能时 需要想要添加什么功能时考虑的东西

添加设备 ?



i 特别说明： DeviceName 可以为空，当为空时，阿里云会颁发产品下的唯一标识符作为 DeviceName。

产品

dawdw

DeviceName ?

请输入 DeviceName

备注名称 ?

请输入备注名称

确认

取消

DeviceName 可以随便填一个 但最好能够符合产品的功能

实例详情

设备管理

产品

设备

分组

任务

数字孪生 New

规则引擎

监控运维

设备划归

安全中心

仿真实验 New

文档与工具

物联网平台 / 设备管理 / 设备 / 设备详情

← dwadf 未激活

产品 dwdw 查看

ProductKey hih0VhtvDls 复制

DeviceSecret ***** 查看

设备信息

Topic 列表

物模型数据

设备影子

文件管理

日志服务

在线调试

分组

任务

设备信息

产品名称	dwdw	ProductKey	hih0VhtvDls 复制
节点类型	设备	DeviceName	dwadf 复制
备注名称	编辑	IP地址	-
创建时间	2022/07/26 19:18:21	激活时间	-
当前状态	未激活	实时延迟	测试
MQTT 连接参数	查看		

设备的信息大家都可以在这里看到

特别注意 MQTT 的连接参数 下面的数据上传时需要用到

这一部分是我创建的新的设备的 MQTT 连接参数

MQTT 连接参数

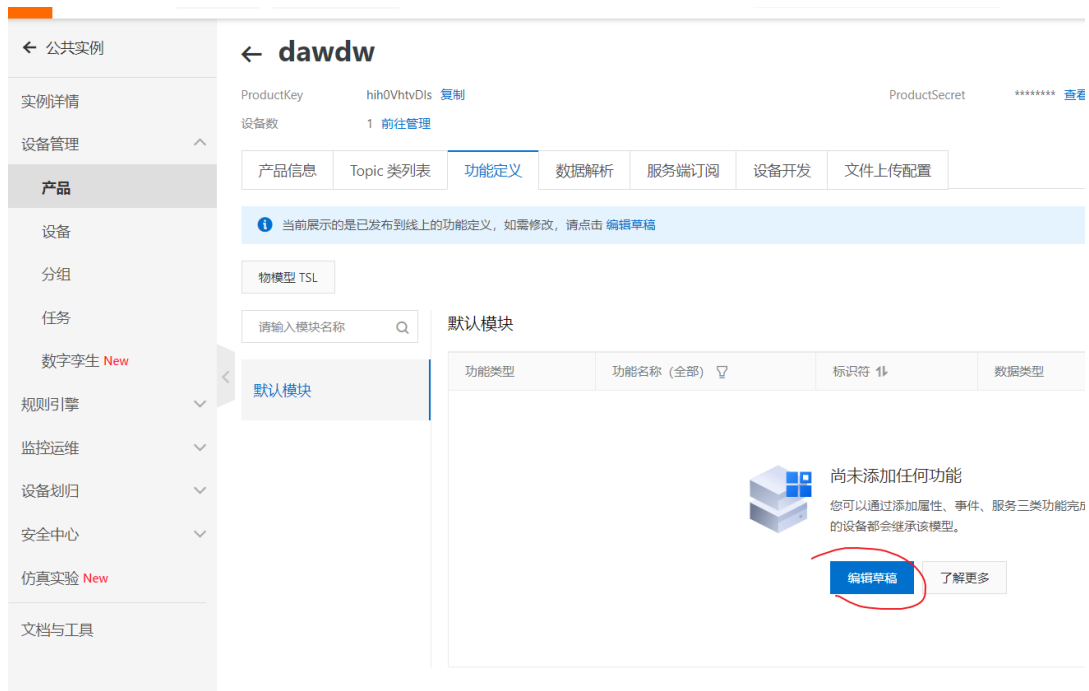
×

clientId	hih0VhtvDls.dwadf securemode=2,signmethod=hmacsha256,timestamp=1658834310673
username	dwadf&hih0VhtvDls
passwd	c69511217a482d3f0b8247d0f9dc40b42cefba849bb25a31aeaf6e2ef280708c
mqttHostUrl	iot-06z00cvdueerv4n.mqtt.iothub.aliyuncs.com
port	1883

一键复制

关闭

为产品添加自定义功能



点进去之后，选择添加自定义功能，根据自己的需要添加功能。我们的项目只需要将下位机的温度数据传到上位机上面实时显示就可以了 所以这里选择属性就ok，这里展示一个温度的 湿度是一样的。

添加自定义功能

×

* 功能类型 ?

属性

服务

事件

* 功能名称 ?

温度

* 标识符 ?

CurrentTemperature

* 数据类型

int32 (整数型) ▼

取值范围

最小值

 ~

最大值

步长

请输入步长

单位

请选择单位 ▼

* 读写类型

☐ 读写

☒ 只读

描述

温度

2/100

这里功能部分的定义就结束了。下面就是上传和程序这一块了。

先放上源代码 下面用截图来说明

```
from umqtt.simple import MQTTClient
from machine import Pin
import network
import time
import machine
```

```

import dht
from machine import Timer

SSID="nova 7 5G"
PASSWORD="1234567890.."

SERVER ='iot-06z00cvdueerv4n.mqtt.iothub.aliyuncs.com'#这里使用域名一直连接不上，只能转换为 IP 使用了#端口号，库里面默认使用处理

CLIENT_ID =
"hih0SLqDqQv.umqtt_cccccccccc|securemode=2,signmethod=hmacsha256,timesta
mp=1658824219747|" #设备 ID

username="umqtt_cccccccccc&hih0SLqDqQv"
password="059f116ced3b07530f910bbfb09756a054106d26219d5acb2946a4a7e8b9e
264"

publish_TOPIC = '/sys/hih0SLqDqQv/umqtt_cccccccccc/thing/event/property/post'
subscribe_TOPIC
=' /sys/hih0SLqDqQv/umqtt_cccccccccc/thing/event/property/post_reply'

client=None
mydht=None
def sub_cb(topic, msg):
    print((topic, msg))

def ConnectWifi(ssid,passwd):
    global wlan
    wlan=network.WLAN(network.STA_IF) #create a wlan object
    wlan.active(True) #Activate the network interface
    wlan.disconnect() #Disconnect the last connected
    WiFi
    wlan.connect(ssid,passwd) #connect wifi
    while(wlan.ifconfig()[0]!='0.0.0.0'):
        time.sleep(1)
    print(wlan.ifconfig())

def apptimerevent(mytimer):
    try:
        sensordata=ReadTemHum()

mymessage='{"params":{"CurrentTemperature":%d,"CurrentHumidity":%d},"method
":"thing.event.property.post"}'%(sensordata[0],sensordata[1])

```



```

        client.publish(topic=publish_TOPIC,msg= mymessage, retain=False, qos=0)
except Exception as ex_results2:
    print('exception',ex_results2)
    mytimer.deinit()
#    finally:
#        machine.reset()

def ReadTemHum():
    mydht.measure()
    tem=mydht.temperature()
    hum=mydht.humidity()
    data=[tem,hum]
    print(data)

    return data

if __name__=='__main__':
#    try:
        mydht=dht.DHT11(machine.Pin(4))
        ConnectWifi(SSID,PASSWORD)
        client = MQTTClient(CLIENT_ID, SERVER,0,username,password,60)
#create a mqtt client
        print(client)
        client.set_callback(sub_cb)                #set callback
        client.connect()                            #connect mqtt
        client.subscribe(subscribe_TOPIC)           #client subscribes to a
topic
        mytimer=Timer(0)
        mytimer.init(mode=Timer.PERIODIC, period=5000,callback=apptimerevent)
        while True:
            client.wait_msg()                        #wait message

#    except Exception as ex_results:
#        print('exception1',ex_results)
#    finally:
#        if(client is not None):
#            client.disconnect()
#        wlan.disconnect()
#        wlan.active(False)
#

```

关于使用 MQTTx 进行虚拟访问阿里云的方法, 在原 CSDN 博客里面介绍的十分详细, 由于这个教程是针对用 esp32 进行的下位机访问云端教程, 这里不再赘述。

上面的代码有一些地方需要你大家根据自己的设定进行改动。

```
SSID="nova 7 5G"
PASSWORD="1234567890.."

SERVER = 'iot-06z00cvdueerv4n.mqtt.iothub.aliyuncs.com'|

CLIENT_ID = "hih0SLqDqQv.umqtt_cccccccccccc|securemode=2,signmethod=hmacsha256,timestamp=1658824219747|" #设备ID
username="umqtt_cccccccccccc&hih0SLqDqQv"
password="059f116ced3b07530f910bbfb09756a054106d26219d5acb2946a4a7e8b9e264"

publish_TOPIC = '/sys/hih0SLqDqQv/umqtt_cccccccccccc/thing/event/property/post'
subscribe_TOPIC = '/sys/hih0SLqDqQv/umqtt_cccccccccccc/thing/event/property/post_reply'
```

首先是这一部分。SSID 和 PASSWORD 是我手机热点的编号和密码, 如果有 WiFi 用 WiFi 的编号和密码也可以。这一部分是 esp32 去连接局域网的时候需要用的。

特别注意：esp32 和电脑必须使用在同一个局域网之下，不然是不能用的（这个程序）

对着之前说过的 MQTT 的连接参数

mqttHostUrl->SERVER

clientId->CLIENT_ID

username->username

passwd->password

MQTT 连接参数

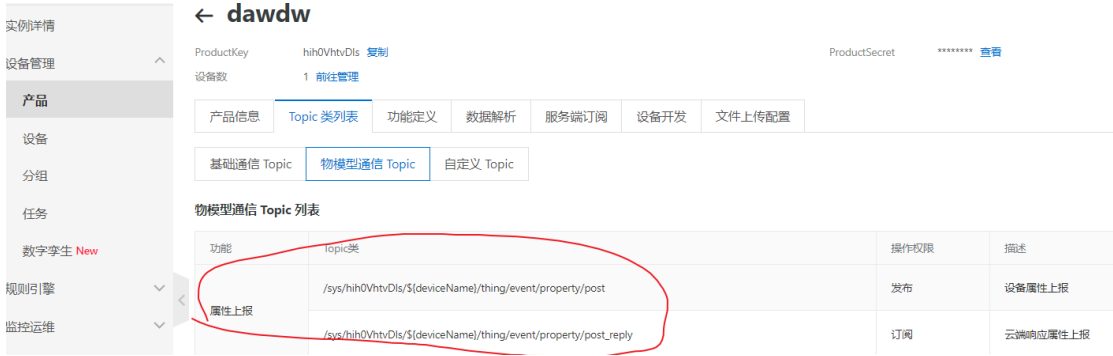
clientId	hih0SLqDqQv.umqtt_cccccccccccc securemode=2,signmethod=hmacsha256,timestamp=1658835478096
username	umqtt_cccccccccccc&hih0SLqDqQv
passwd	43082b171091ad9a0dd0d20049fc6fe781165f557c097d9616ea24041dc405ef
mqttHostUrl	iot-06z00cvdueerv4n.mqtt.iothub.aliyuncs.com
port	1883

一键复制

关闭

按照这几个参数在程序中配置好

同时，按照自定义功能的类型，我们在程序中需要把属性上报的请求头按照物理层通信的标准请求头写好



在程序中对应该一部分

```
publish_TOPIC = '/sys/hih0SLqDqQv/umqtt_ccccccccccc/thing/event/property/post'
subscribe_TOPIC = '/sys/hih0SLqDqQv/umqtt_ccccccccccc/thing/event/property/post_reply'
```

下面是数据传输部分

```
def apptimerevent(mytimer):
    try:
        sensordata=ReadTemHum()
        mymessage='{"params":{"CurrentTemperature":%d,"CurrentHumidity":%d},"method":"thing.event.property.post"}'
        %(sensordata[0],sensordata[1])
        client.publish(topic=publish_TOPIC,msg= mymessage, retain=False, qos=0)
    except Exception as ex_results2:
        print('exception',ex_results2)
        mytimer.deinit()
    # finally:
    ...
```

这一部分是定时向云端发送数据。

参数这一部分务必与之前自定义功能的标识符相匹配，否则数据传上去会识别不到。

添加自定义功能



* 功能类型 ?

属性	服务	事件
----	----	----

* 功能名称 ?

* 标识符 ?

Client.publish()这个函数是用来做传输的 两个参数分别是之前定义好的请求头 topic 和要发向云端的数据。

主循环这一部分

```
if __name__ == '__main__':
    # try:
        mydht=dht.DHT11(machine.Pin(4))
        ConnectWifi(SSID,PASSWORD)
        client = MQTTClient(CLIENT_ID, SERVER,0,username,password,60)    #create a mqtt clier
        print(client)
        client.set_callback(sub_cb)                                       #set callback
        client.connect()                                                  #connect mqtt
        client.subscribe(subscribe_TOPIC)|                                #client subscribes to a topic
        mytimer=Timer(0)
        mytimer.init(mode=Timer.PERIODIC, period=5000,callback=apptimerevent)
        while True:
            client.wait_msg()                                             #wait message
```

这一部分包含了初始化 MQTT 客户端，esp32 连接局域网，以及一个定时进行通信请求的定时器。

Client.subscribe()是基于请求头对阿里云端进行订阅，这样才可以让 esp32 收到来自云端服务器的反馈。

```
{b'/sys/hih0SLqDqQv/umqtt_ccccccccccc/thing/event/property/post_reply', b'{"code":200,"data":{},"id":"null","message":"success","method":"thing.event.property.post","version":"1.0"}'}
[29, 64]
```

当正常通信开始后，服务器定时进行反馈，触发订阅中断，上面这部分是订阅中断显示的信息。当服务器代码为 200 时，通信时是正常的。



服务器与 esp32 的实时通信可以在这里面查到。状态为 200，绿色时是正常的 红色代表出问题了，可以根据错误代码类型进行 debug。

一轮通信分为设备到云->物理模型->物理模型处理->云反馈回设备这一整个过程

2022/07/26 20:00:39.948	0a3062fb16588368399272...	15519003035...	查看	umqtt_ccccccccccc	云到设备消息	/sys/hih0SLqDqQv/...	{"Params":{"null":"Result..."	200
2022/07/26 20:00:39.943	0a3062fb16588368399272...	-	-	umqtt_ccccccccccc	物模型	Check	{"Params":{"CurrentHu...	200
2022/07/26 20:00:39.940	0a3062fb16588368399272...	15519003035...	查看	umqtt_ccccccccccc	物模型消息	/sys/hih0SLqDqQv/...	{"Requestid":"","Co...	200
2022/07/26 20:00:39.931	0a3062fb16588368399272...	15519003035...	查看	umqtt_ccccccccccc	设备到云消息	/sys/hih0SLqDqQv/...	{"Params":{"null":"Result..."	200

成功连接之后，就可以看到 esp32 传输过来的数据了。



同时，阿里云是支持同时在线多个设备的。可以进行多个设备的实时监控。

设备

全部产品

设备总数 5

激活设备 4

当前在线 1

设备列表

批次管理

高级搜索

添加设备

批量添加

DeviceName

请输入 DeviceName

请选择设备标签

<input type="checkbox"/>	DeviceName/备注名称	设备所属产品	节点类型	状态/启用状态	最后上报时间
<input type="checkbox"/>	dwardf	dawdw	设备	未激活	-
<input type="checkbox"/>	umqtt_ccccccccccc	umqtt_ccc	设备	在线	2022/07/26 19:43:49.285
<input type="checkbox"/>	ESP32-MQTT2	MQTT-device	设备	离线	2022/07/26 14:47:29.970
<input type="checkbox"/>	ESP32-MQTT	MQTT-device	设备	离线	2022/07/26 15:23:35.347
<input type="checkbox"/>	device1	my-mqtt-device	设备	离线	2022/07/26 15:49:39.839