**Node JS直接使用MQTT协议调用直接方法**

Azure IOT SDK中封装了使用MQTT协议调用直接方法的用法，如果想直接使用MQTT协议自己封装调用Azure IOT来实现直接方法的调用。虽然官方文档也有介绍，但依照官方文档的步骤，依然无法调用，这是因为文档中讲解的一些参数细节并不详细，具体可以参考以下文档和代码

使用NodeJS调用MQTT协议，需要依赖MQTT.JS ，该项目主页：<https://github.com/mqttjs/MQTT.js>

首先建立NodeJS项目目录，然后安装MQTT依赖包

npm install mqtt --save

其次，在代码中配置连接IOT字符串，连接并接受云端发送调用直接方法消息

var mqtt = require('mqtt')

var URL = require('url');

var QueryString = require('querystring');

var options = {

cmd: 'connect',

protocolId: 'MQTT',

protocolVersion: 4,

clean: false,

clientId: 'kevindevice',

rejectUnauthorized: true,

username: 'xuhuaIOT.azure-devices.cn/kevindevice/api-version=2016-11-14',

reconnectPeriod: 0, // Client will handle reconnection at the higher level

password : new Buffer('SharedAccessSignature sr=xuhuaIOT.azure-devices.cn%2Fdevices%2Fkevindevice&sig=wU3c8uu6AqTKoKF0OAtBE3K7ZcAkhelhFux6vNYEiho%3D&se=1485411010')

};

var client = mqtt.connect('mqtts://xuhuaIOT.azure-devices.cn', options);

client.on('connect', () => {

client.subscribe('$iothub/methods/POST/#', { qos: 0 }, function (err) {

if (err) {

console.error("Direct method error: " + err.message);

} else {

console.log("Successfully subscribe to direct method");

}

});

})

client.on('message', (topic, message) => {

console.log(topic);

var methodMessage = parseMessage(topic, message);

console.log('message' + methodMessage);

client.publish('$iothub/methods/res/200/?$rid=' + methodMessage.requestId, JSON.stringify({"message":"reboot successfully"}), { qos: 0, retain: false }, function(err) {});

})

var handleRebootRequest = function(message) {

console.log('Response to method \'' + message + '\' sent successfully.');

}

var parseMessage = function(topic, body) {

var url, path, query;

try {

url = URL.parse(topic);

path = url.path.split('/');

query = QueryString.parse(url.query);

}

catch(err) {

console.error(err)

return undefined;

}

// if the topic has a querystring then 'path' will include it; so

// we strip it out

var lastPathComponent = path[path.length - 1];

if(lastPathComponent.indexOf('?') !== -1) {

path[path.length - 1] = lastPathComponent.substr(

0, lastPathComponent.indexOf('?')

);

}

if(path.length > 0 && path[0] === '$iothub') {

var message = {};

if(path.length > 1 && path[1].length > 0) {

// create an object for the module; for example, $iothub/twin/...

// would result in there being a message.twin object

var mod = message[path[1]] = {};

// parse the request ID if there is one

if(!!(query.$rid)) {

message.requestId = query.$rid;

}

// parse the other properties properties (excluding $rid)

message.properties = query;

delete message.properties.$rid;

// save the body

message.body = body;

// parse the verb

if(path.length > 2 && path[2].length > 0) {

mod.verb = path[2];

// This is a topic that looks like this:

// $iothub/methods/POST/{method name}?$rid={request id}&{serialized properties}

// We parse the method name out.

if(path.length > 3 && path[3].length > 0) {

mod.methodName = path[3];

} else {

// The service published a message on a strange topic name. This is

// probably a service bug. At any rate we don't know what to do with

// this strange topic so we throw.

throw new Error('Device method call\'s MQTT topic name does not include the method name.');

}

}

}

return message;

}

return undefined;

}

本示例是设备接受Azure IOTHub发来的调用直接方法消息，并进行响应处理。

注意：IOT的用户名必需是{iothubhostname}/{device\_id}/ api-version={api-version}, 官方文档中并没有提及要记入api-version，但实际测试下来，这里是必须要有的。

参考文档：

<https://docs.microsoft.com/zh-cn/azure/iot-hub/iot-hub-mqtt-support#a-nameusing-the-mqtt-protocol-directlya直接使用-mqtt-协议>

<https://docs.microsoft.com/zh-cn/azure/iot-hub/iot-hub-devguide-direct-methods#a-namehandle-a-direct-method-on-a-devicea在设备上处理直接方法>