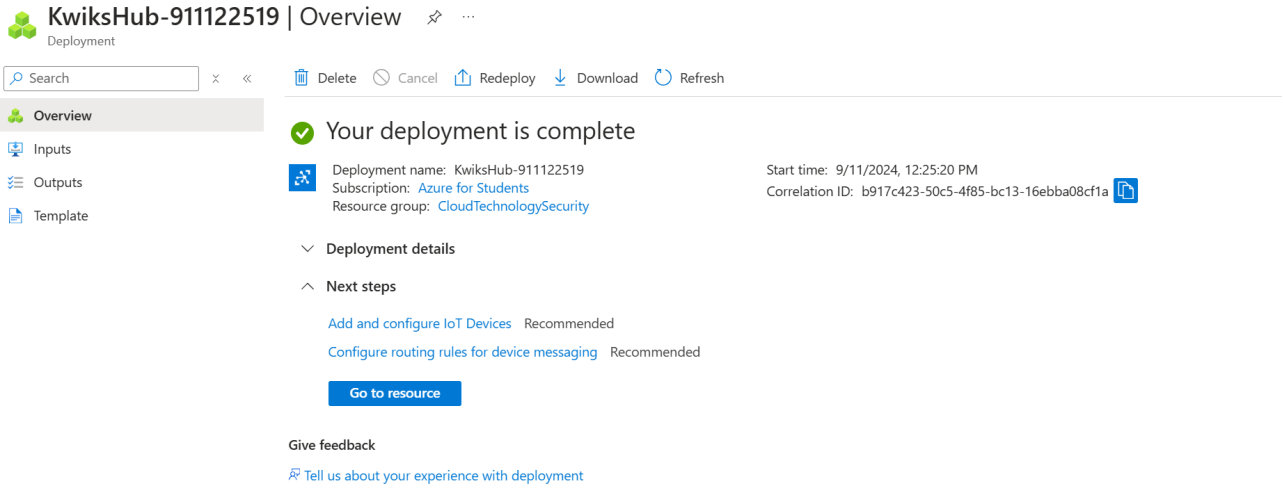
Opdracht 3 Cloud Technology & Security

Qing Scholten (20208294)

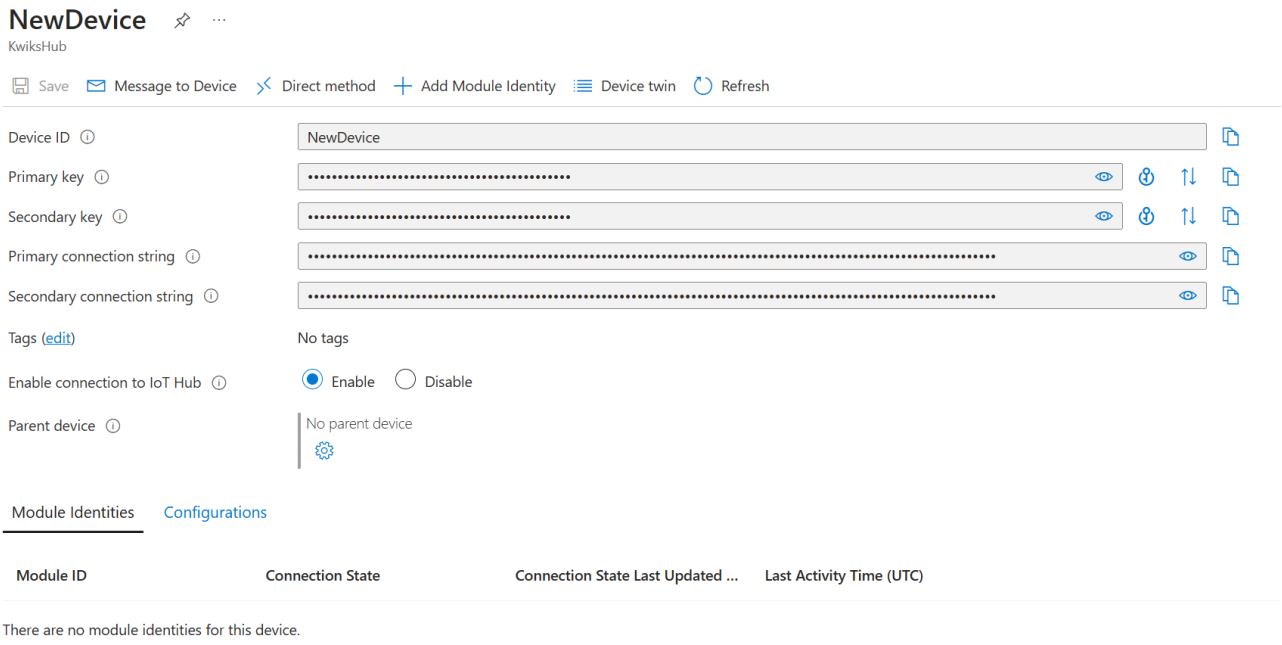
Voorbereiding

1. De IOT Hub is succesvol aangemaakt.



Practicum opgave

1. De device is succesvol aangemaakt.



3. az iot hub monitor-events --output table -p all -n KwiksHub

```
Starting event monitor, use ctrl-c to stop...
event:
  annotations:
    iothub-connection-auth-generation-id: '638616475937603942'
    iothub-connection-auth-method: '{"scope":"device","type":"sas","issuer":"iothub","acceptingIpFilterRule":null}'
    iothub-connection-device-id: NewDevice
    iothub-enqueuedtime: 1726051793416
    iothub-message-source: Telemetry
    x-opt-enqueued-time: 1726051793519
    x-opt-offset: '2080'
    x-opt-sequence-number: 4
    x-opt-sequence-number-epoch: -1
  component: ''
  interface: ''
  module: ''
  origin: NewDevice
  payload: '{"deviceId":"myFirstDevice","windSpeed":10.711730036851169,"temperature":24.203650171091322,"humidity":64.7160756057007}'
  properties:
    application:
      temperatureAlert: 'false'

event:
  annotations:
    iothub-connection-auth-generation-id: '638616475937603942'
    iothub-connection-auth-method: '{"scope":"device","type":"sas","issuer":"iothub","acceptingIpFilterRule":null}'
    iothub-connection-device-id: NewDevice
    iothub-enqueuedtime: 1726051795414
    iothub-message-source: Telemetry
    x-opt-enqueued-time: 1726051795425
    x-opt-offset: '2600'
    x-opt-sequence-number: 5
    x-opt-sequence-number-epoch: -1
  component: ''
  interface: ''
  module: ''
  origin: NewDevice
  payload: '{"deviceId":"myFirstDevice","windSpeed":12.331555729678886,"temperature":24.657019220139656,"humidity":60.535278067455764}'
  properties:
    application:
      temperatureAlert: 'false'
```

```
event:
  annotations:
    iothub-connection-auth-generation-id: '638616475937603942'
    iothub-connection-auth-method: '{"scope":"device","type":"sas","issuer":"iothub","acceptingIpFilterRule":null}'
    iothub-connection-device-id: NewDevice
    iothub-enqueuedtime: 1726051795414
    iothub-message-source: Telemetry
    x-opt-enqueued-time: 1726051795425
    x-opt-offset: '2600'
    x-opt-sequence-number: 5
    x-opt-sequence-number-epoch: -1
  component: ''
  interface: ''
  module: ''
  origin: NewDevice
  payload: '{"deviceId":"myFirstDevice","windSpeed":12.331555729678886,"temperature":24.657019220139656,"humidity":60.535278067455764}'
  properties:
    application:
      temperatureAlert: 'false'

event:
  annotations:
    iothub-connection-auth-generation-id: '638616475937603942'
    iothub-connection-auth-method: '{"scope":"device","type":"sas","issuer":"iothub","acceptingIpFilterRule":null}'
    iothub-connection-device-id: NewDevice
    iothub-enqueuedtime: 1726051797414
    iothub-message-source: Telemetry
    x-opt-enqueued-time: 1726051797425
    x-opt-offset: '3128'
    x-opt-sequence-number: 6
    x-opt-sequence-number-epoch: -1
  component: ''
  interface: ''
  module: ''
  origin: NewDevice
  payload: '{"deviceId":"myFirstDevice","windSpeed":10.44689103489618,"temperature":21.027827263166486,"humidity":70.46616199548718}'
  properties:
    application:
      temperatureAlert: 'false'
```

```
  component: ''
  interface: ''
  module: ''
  origin: NewDevice
  payload: '{"deviceId":"myFirstDevice","windSpeed":12.331555729678886,"temperature":24.657019220139656,"humidity":60.535278067455764}'
  properties:
    application:
      temperatureAlert: 'false'

event:
  annotations:
    iothub-connection-auth-generation-id: '638616475937603942'
    iothub-connection-auth-method: '{"scope":"device","type":"sas","issuer":"iothub","acceptingIpFilterRule":null}'
    iothub-connection-device-id: NewDevice
    iothub-enqueuedtime: 1726051797414
    iothub-message-source: Telemetry
    x-opt-enqueued-time: 1726051797425
    x-opt-offset: '3128'
    x-opt-sequence-number: 6
    x-opt-sequence-number-epoch: -1
  component: ''
  interface: ''
  module: ''
  origin: NewDevice
  payload: '{"deviceId":"myFirstDevice","windSpeed":10.44689103489618,"temperature":21.027827263166486,"humidity":70.46616199548718}'
  properties:
    application:
      temperatureAlert: 'false'

event:
  annotations:
    iothub-connection-auth-generation-id: '638616475937603942'
    iothub-connection-auth-method: '{"scope":"device","type":"sas","issuer":"iothub","acceptingIpFilterRule":null}'
    iothub-connection-device-id: NewDevice
    iothub-enqueuedtime: 1726051799430
    iothub-message-source: Telemetry
    x-opt-enqueued-time: 1726051799425
    x-opt-offset: '3648'
    x-opt-sequence-number: 7
    x-opt-sequence-number-epoch: -1
  component: ''
  interface: ''
```


Het bericht is afkomstig van NewDevice. Dit is consistent.


```
PS C:\Users\Kwik\Documents\HBO-ICT\IOT\Cloud Technology & Security> node .\NewDevice.js
Client connected
Sending message: {"deviceId":"myFirstDevice","windSpeed":10.711730036851169,"temperature":24.203650171091322,"humidity":64.7160756057007}
send status: MessageEnqueued
Sending message: {"deviceId":"myFirstDevice","windSpeed":12.331555729678886,"temperature":24.657019220139656,"humidity":60.535278067455764}
send status: MessageEnqueued
Sending message: {"deviceId":"myFirstDevice","windSpeed":10.44689103489618,"temperature":21.027827263166486,"humidity":70.46616199548718}
send status: MessageEnqueued
Sending message: {"deviceId":"myFirstDevice","windSpeed":12.3739032900647,"temperature":23.5720554432253,"humidity":75.36036975694071}
send status: MessageEnqueued
PS C:\Users\Kwik\Documents\HBO-ICT\IOT\Cloud Technology & Security>
```


---

4. De vier events zijn 'connect', 'error', 'disconnect' en 'message'. Deze vier hebben een handler in het NewDevice voorbeeld. In de vorige vraag is het 'connect' event gebruikt.

5.

```
Starting event monitor, use ctrl-c to stop...
event:
  annotations:
    iothub-connection-auth-generation-id: '638616475937603942'
    iothub-connection-auth-method: '{"scope":"device","type":"sas","issuer":"iothub","acceptingIpFilterRule":null}'
    iothub-connection-device-id: NewDevice
    iothub-enqueuedtime: 1726053006928
    iothub-message-source: Telemetry
    x-opt-enqueued-time: 1726053007024
    x-opt-offset: '5840'
    x-opt-sequence-number: 11
    x-opt-sequence-number-epoch: -1
  component: ''
  interface: ''
  module: ''
  origin: NewDevice
  payload: '{"deviceId":"myFirstDevice","windSpeed":13.224960276029325,"temperature":26.82384440111823,"humidity":64.71812811394274,"time":"6:52"}'
  properties:
    application:
      temperatureAlert: 'false'

event:
  annotations:
    iothub-connection-auth-generation-id: '638616475937603942'
    iothub-connection-auth-method: '{"scope":"device","type":"sas","issuer":"iothub","acceptingIpFilterRule":null}'
    iothub-connection-device-id: NewDevice
    iothub-enqueuedtime: 1726053008929
    iothub-message-source: Telemetry
    x-opt-enqueued-time: 1726053008931
    x-opt-offset: '6376'
    x-opt-sequence-number: 12
    x-opt-sequence-number-epoch: -1
  component: ''
  interface: ''
  module: ''
  origin: NewDevice
  payload: '{"deviceId":"myFirstDevice","windSpeed":10.833432949369492,"temperature":26.975442637427385,"humidity":64.39638316176219,"time":"22:56"}'
  properties:
    application:
      temperatureAlert: 'false'
```

```
PS C:\Users\Kwik\Documents\HBO-ICT\IOT\Cloud Technology & Security> node .\NewDevice.js
Client connected
Sending message: {"deviceId":"myFirstDevice","windSpeed":13.224960276029325,"temperature":26.82384440111823,"humidity":64.71812811394274,"time":"6:52"}
send status: MessageEnqueued
Sending message: {"deviceId":"myFirstDevice","windSpeed":10.833432949369492,"temperature":26.975442637427385,"humidity":64.39638316176219,"time":"22:56"}
send status: MessageEnqueued
PS C:\Users\Kwik\Documents\HBO-ICT\IOT\Cloud Technology & Security>
```

```
function generateMessage() {
  const windSpeed = 10 + (Math.random() * 4); // range: [10, 14]
  const temperature = 20 + (Math.random() * 10); // range: [20, 30]
  const humidity = 60 + (Math.random() * 20); // range: [60, 80]
  const time = Math.floor(Math.random()*23) + ":" + Math.floor(Math.random()*59);
  const data = JSON.stringify({ deviceId: 'myFirstDevice', windSpeed: windSpeed, temperature: temperature, humidity: humidity, time: time });
  const message = new Message(data);
  message.properties.add('temperatureAlert', (temperature > 28) ? 'true' : 'false');
  return message;
}
```

6.

```
function generateMessage() {
  const temperature = 20 + (Math.random() * 10); // range: [20, 30]
  const data = JSON.stringify({ temperature: temperature });
  const message = new Message(data);
  message.properties.add('temperatureAlert', (temperature > 28) ? 'true' : 'false');
  return message;
}
```

```
PS C:\Users\Kwik\Documents\HBO-ICT\IOT\Cloud Technology & Security> node .\NewSensor.js
Client connected
Sending message: {"temperature":27.521829947749147}
send status: MessageEnqueued
Sending message: {"temperature":24.75442414473207}
send status: MessageEnqueued
Sending message: {"temperature":25.19484310303998}
send status: MessageEnqueued
PS C:\Users\Kwik\Documents\HBO-ICT\IOT\Cloud Technology & Security>
```

```
PS /home/qing> az iot hub monitor-events --output table -p all -n KwiksHub
Starting event monitor, use ctrl-c to stop...
event:
  annotations:
    iothub-connection-auth-generation-id: '638616475937603942'
    iothub-connection-auth-method: '{"scope":"device","type":"sas","issuer":"iothub","acceptingIpFilterRule":null}'
    iothub-connection-device-id: NewDevice
    iothub-enqueuedtime: 1726053415686
    iothub-message-source: Telemetry
    x-opt-enqueued-time: 1726053415767
    x-opt-offset: '6912'
    x-opt-sequence-number: 13
    x-opt-sequence-number-epoch: -1
  component: ''
  interface: ''
  module: ''
  origin: NewDevice
  payload: '{"temperature":27.521829947749147}'
  properties:
    application:
      temperatureAlert: 'false'

event:
  annotations:
    iothub-connection-auth-generation-id: '638616475937603942'
    iothub-connection-auth-method: '{"scope":"device","type":"sas","issuer":"iothub","acceptingIpFilterRule":null}'
    iothub-connection-device-id: NewDevice
    iothub-enqueuedtime: 1726053417686
    iothub-message-source: Telemetry
    x-opt-enqueued-time: 1726053417689
    x-opt-offset: '7352'
    x-opt-sequence-number: 14
    x-opt-sequence-number-epoch: -1
  component: ''
  interface: ''
  module: ''
  origin: NewDevice
  payload: '{"temperature":24.75442414473207}'
  properties:
    application:
      temperatureAlert: 'false'
```

```
event:
  annotations:
    iothub-connection-auth-generation-id: '638616475937603942'
    iothub-connection-auth-method: '{"scope":"device","type":"sas","issuer":"iothub","acceptingIpFilterRule":null}'
    iothub-connection-device-id: NewDevice
    iothub-enqueuedtime: 1726053419686
    iothub-message-source: Telemetry
    x-opt-enqueued-time: 1726053419689
    x-opt-offset: '7784'
    x-opt-sequence-number: 15
    x-opt-sequence-number-epoch: -1
  component: ''
  interface: ''
  module: ''
  origin: NewDevice
  payload: '{"temperature":25.19484310303998}'
  properties:
    application:
      temperatureAlert: 'false'
```

7.

```
function generateMessage() {
  const windSpeed = 10 + (Math.random() * 4); // range: [10, 14]
  const temperature = 20 + (Math.random() * 10); // range: [20, 30]
  const humidity = 60 + (Math.random() * 20); // range: [60, 80]
  const time = Math.floor(Math.random()*23) + ":" + Math.floor(Math.random()*59);
  const data = JSON.stringify({ deviceId: 'myFirstDevice', windSpeed: windSpeed, temperature: temperature, humidity: humidity, time: time });
  const message = new Message(data);
  message.properties.add('temperatureAlert', (temperature > 28) ? 'true' : 'false');
  message.contentType = 'application/json'
  return message;
}
```

```
PS C:\Users\Kwik\Documents\HBO-ICT\IOT\Cloud Technology & Security> node .\NewDevice.js
Client connected
Sending message: {"deviceId":"myFirstDevice","windSpeed":12.217805672536812,"temperature":29.425868540754937,"humidity":77.17309154035989,"time":"0:40"}
send status: MessageEnqueued
PS C:\Users\Kwik\Documents\HBO-ICT\IOT\Cloud Technology & Security>
```

```
PS /home/qing> az iot hub monitor-events --output table -p all -n KwiksHub
Starting event monitor, use ctrl-c to stop...
event:
  annotations:
    iothub-connection-auth-generation-id: '638616475937603942'
    iothub-connection-auth-method: '{"scope":"device","type":"sas","issuer":"iothub","acceptingIpFilterRule":null}'
    iothub-connection-device-id: NewDevice
    iothub-enqueuedtime: 1726054329340
    iothub-message-source: Telemetry
    x-opt-enqueued-time: 1726054329405
    x-opt-offset: '9872'
    x-opt-sequence-number: 19
    x-opt-sequence-number-epoch: -1
  component: ''
  interface: ''
  module: ''
  origin: NewDevice
  payload:
    deviceId: myFirstDevice
    humidity: 77.17309154035989
    temperature: 29.425868540754937
    time: 0:40
    windSpeed: 12.217805672536812
  properties:
    application:
      temperatureAlert: 'true'
    system:
      content_type: application/json
```

---

9. De methodes "getDeviceLog" en "lockDoor".

10.

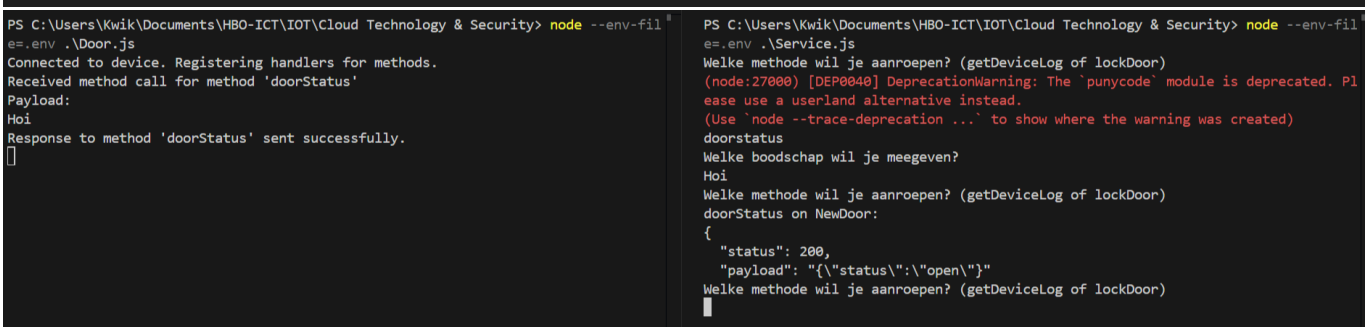
```
// Copyright (c) Microsoft. All rights reserved.
// Licensed under the MIT license. See LICENSE file in the project root for full license information.

'use strict';

var Client = require('azure-iothub').Client;
const readline = require('readline');
const rl = readline.createInterface({
  input: process.stdin,
  output: process.stdout
});

var connectionString = process.env.IOTHUB_CONNECTION_STRING;
var targetDevice = "NewDoor";
var client = Client.fromConnectionString(connectionString);

var send = 0;
var methode = "";
var payload = "";
rl.setPrompt(`Welke methode wil je aanroepen? (getDeviceLog of lockDoor)\n`);
rl.prompt();
rl.on('line', (input) => {
  if (send == 0) {
      methode = input.toLowerCase();
      if (methode == 'getdevicelog') {
          methode = "getDeviceLog";
      }
      else if (methode == 'lockdoor') {
        methode = 'lockDoor';
      }
      rl.setPrompt("Welke boodschap wil je meegeven?\n");
      rl.prompt();
      send = 1;
  }
  else if (send == 1) {
      payload = input;
      rl.setPrompt("Welke methode wil je aanroepen? (getDeviceLog of lockDoor)\n");
      rl.prompt();
      send = 0;
      var methodParams = {
        methodName: methode,
        payload: payload,
        responseTimeoutInSeconds: 15
      };
      client.invokeDeviceMethod(targetDevice, methodParams, function (err, result) {
        if (err) {
          console.error('Failed to invoke method \'' + methodParams.methodName + '\': ' + err.message);
        } else {
          console.log(methodParams.methodName + ' on ' + targetDevice + ':');
          console.log(JSON.stringify(result, null, 2));
        }
      });
  }
});
```

```
PS C:\Users\Kwik\Documents\HBO-ICT\IOT\Cloud Technology & Security> node --env-file=.env .\Door.js
Connected to device. Registering handlers for methods.
Received method call for method 'getDeviceLog'
Payload:
Hello World
Response to method 'getDeviceLog' sent successfully.
Received method call for method 'lockDoor'
Payload:
Hello World 2
Response to method 'lockDoor' sent successfully.
```

```
PS C:\Users\Kwik\Documents\HBO-ICT\IOT\Cloud Technology & Security> node --env-file=.env .\Service.js
Welke methode wil je aanroepen? (getDeviceLog of lockDoor)
(node:1716) [DEP0040] DeprecationWarning: The `punycode` module is deprecated. Please use a userland alternative instead.
(Use `node --trace-deprecation ...` to show where the warning was created)
getdevicelog
Welke boodschap wil je meegeven?
Hello World
Welke methode wil je aanroepen? (getDeviceLog of lockDoor)
getDeviceLog on NewDoor:
{
  "status": 200,
  "payload": "example payload"
}
lockdoor
Welke boodschap wil je meegeven?
```

11.

```
function onDoorStatus(request, response) {
    printDeviceMethodRequest(request);
    var payload = JSON.stringify({ status: deurstatus })
    response.send(200, payload, function(err) {
        if (err) {
            console.error('An error ocurred when sending a method response:\n' +
                err.toString());
        }
        else {
            console.log('Response to method \'' + request.methodName +
                '\' sent successfully.' );
        }
    });
}
```

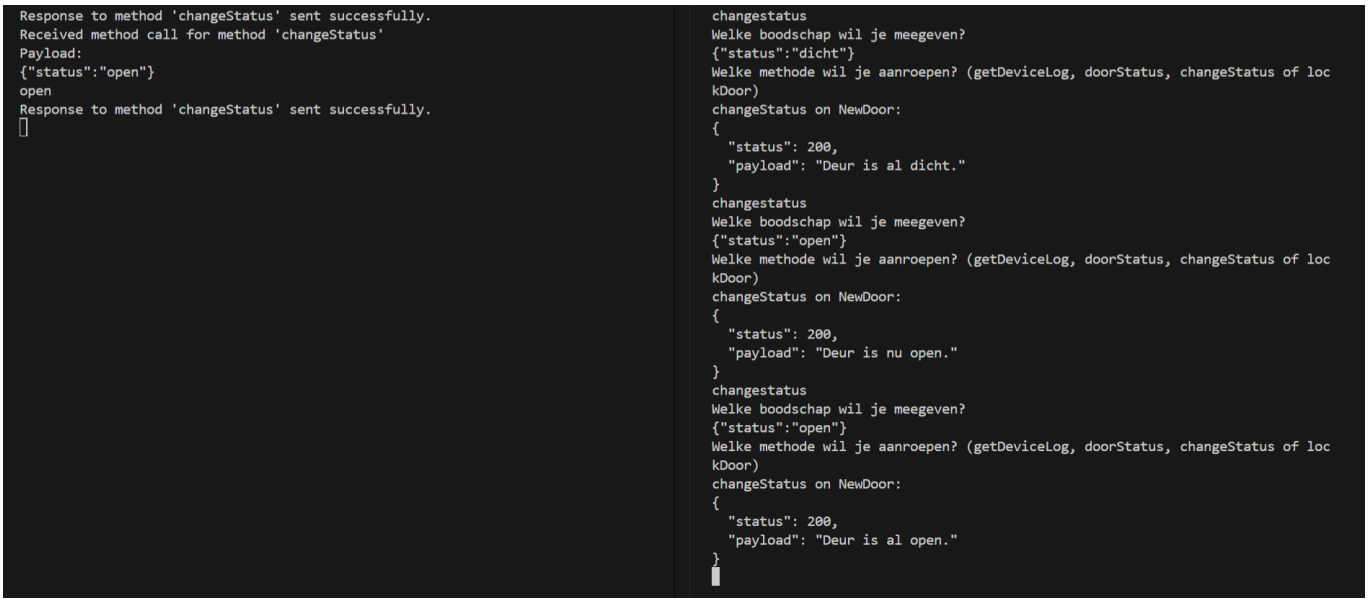


12.

```
function onChangeStatus(request, response) {
    printDeviceMethodRequest(request);
    var newstatus = "";
    var antwoord = "";
    // Implement actual logic here.
    try {
        newstatus = JSON.parse(request.payload);
        console.log(newstatus.status);
    }
    catch (e) {
        console.error('An error ocurred when changing door status:\n' +
            err.toString());
    }
    if (newstatus.status=="open") {
        if (deurstatus == "open") {
            antwoord = "Deur is al open.";
        }
        else {
            antwoord = "Deur is nu open.";
            deurstatus = "open";
        }
    }
    else if (newstatus.status=="dicht") {
        if (deurstatus == "dicht") {
            antwoord = "Deur is al dicht.";
        }
        else {
            antwoord = "Deur is nu dicht.";
            deurstatus = "dicht";
        }
    }
    // complete the response
    response.send(200, antwoord, function (err) {
        if(err) {
            console.error('An error ocurred when sending a method response:\n' +
                err.toString());
        } else {
            console.log('Response to method \'' + request.methodName +
                '\' sent successfully.' );
        }
    });
}
```

```
PS C:\Users\Kwik\Documents\HBO-ICT\IOT\Cloud Technology & Security> node --env-file=.env .\Door.js
Connected to device. Registering handlers for methods.
Received method call for method 'changeStatus'
Payload:
{"status": "dicht"}
dicht
Response to method 'changeStatus' sent successfully.
Received method call for method 'changeStatus'
Payload:
{"status":"dicht"}
dicht
Response to method 'changeStatus' sent successfully.
Received method call for method 'changeStatus'
Payload:
{"status":"open"}
open
```

```
PS C:\Users\Kwik\Documents\HBO-ICT\IOT\Cloud Technology & Security> node --env-file=.env .\Service.js
Welke methode wil je aanroepen? (getDeviceLog, doorStatus, changeStatus of lockDoor)
(node:27068) [DEP0040] DeprecationWarning: The `punycode` module is deprecated. Please use a userland alternative instead.
(Use `node --trace-deprecation ...` to show where the warning was created)
changestatus
Welke boodschap wil je meegeven?
{"status": "dicht"}
Welke methode wil je aanroepen? (getDeviceLog, doorStatus, changeStatus of lockDoor)
changeStatus on NewDoor:
{
  "status": 200,
  "payload": "Deur is nu dicht."
}
```


Response to method 'changeStatus' sent successfully.
Received method call for method 'changeStatus'
Payload:
{"status":"open"}
open
Response to method 'changeStatus' sent successfully.
changestatus
Welke boodschap wil je meegeven?
{"status":"dicht"}
Welke methode wil je aanroepen? (getDeviceLog, doorStatus, changeStatus of lockDoor)
changeStatus on NewDoor:
{
"status": 200,
"payload": "Deur is al dicht."
}
changestatus
Welke boodschap wil je meegeven?
{"status":"open"}
Welke methode wil je aanroepen? (getDeviceLog, doorStatus, changeStatus of lockDoor)
changeStatus on NewDoor:
{
"status": 200,
"payload": "Deur is nu open."
}
changestatus
Welke boodschap wil je meegeven?
{"status":"open"}
Welke methode wil je aanroepen? (getDeviceLog, doorStatus, changeStatus of lockDoor)
changeStatus on NewDoor:
{
"status": 200,
"payload": "Deur is al open."
}

13.

```
'use strict';

const Protocol = require('azure-iot-device-mqtt').Mqtt;

const Client = require('azure-iot-device').Client;
let client = null;
const chalk = require('chalk');
var myinterval = "";

function main() {
    // open a connection to the device
    const deviceConnectionString = process.env.IOTHUB_DEVICE_CONNECTION_STRING_ALARM;
    client = Client.fromConnectionString(deviceConnectionString, Protocol);
    // eslint-disable-next-line security/detect-non-literal-fs-filename
    client.open(onConnect);
}

function onConnect(err) {
    if(err) {
        console.error('Could not connect: ' + err.message);
    } else {
        console.log('Connected to device. Registering handlers for methods.');

        // register handlers for all the method names we are interested in
        client.onDeviceMethod('changeColour', onChangeColour);
        client.onDeviceMethod('changeFrequency', onChangeFrequency);
        process.stdout.write("aan");
        myinterval = setInterval(print, 1000);
    }
}

var toggle = false;
var red = 0;
var green = 255;
var blue = 0;

function print() {
    process.stdout.cursorTo(0);
    if (toggle) {
        process.stdout.write("    ");
    }
    else {
        process.stdout.write(chalk.rgb(red,green,blue)("aan"));
    }
    toggle= (!toggle);
}

function printDeviceMethodRequest(request) {
    // print method name
    console.log('Received method call for method \'' + request.methodName + '\'');

    // if there's a payload just do a default console log on it
    if(request.payload) {
        console.log('Payload:\n' + request.payload);
    }
}

function onChangeColour(request, response) {
    printDeviceMethodRequest(request);
    var check = "";
    if (!isNaN(request.payload.red)) {
        red = request.payload.red;
        check = check + "Rood is veranderd naar " + red + ". ";
    }
    if (!isNaN(request.payload.green)) {
        green = request.payload.green;
        check = check + "Groen is veranderd naar " + green + ". ";
    }
    if (!isNaN(request.payload.blue)) {
        blue = request.payload.blue;
        check = check + "Blauw is veranderd naar " + blue + ". ";
    }
    response.send(200, check, function(err) {
        if (err) {
            console.error('An error occured when sending a method response:\n' +
```

```
                err.toString());
        }
        else {
            console.log('Response to method \'' + request.methodName +
                '\' sent successfully.' );
        }
    });
}

function onChangeFrequency(request, response) {
    printDeviceMethodRequest(request);
    if (!isNaN(request.payload.frequency)) {
        clearInterval(myinterval);
        myinterval = setInterval(print, request.payload.frequency);
    }
    response.send(200, "De frequentie is veranderd naar " + request.payload.frequency, function(err) {
        if (err) {
            console.error('An error ocurred when sending a method response:\n' +
                err.toString());
        }
        else {
            console.log('Response to method \'' + request.methodName +
                '\' sent successfully.' );
        }
    });
}
main();
```

Direct method

NewAlarm

You can use this tool to invoke direct methods on devices from the cloud. Direct methods have a name, payload, and configurable timeouts. Learn more

Device ID

NewAlarm

Method name *

changeColour

Payload

```
{
    "red": "255",
    "green": "0",
    "blue": "0"
}
```

Response timeout: 30 seconds

Connection timeout: Device must already be connected

Invoke method

Result

```
{
    "status": 200,
    "payload": "Rood is veranderd naar 255. Groen is veranderd naar 0. Blauw is veranderd naar 0. "
}
```

```
PS C:\Users\Kwik\Documents\HBO-ICT\IOT\Cloud Technology & Security> node --env-file=.env .\Alarm.js
Connected to device. Registering handlers for methods.
aanReceived method call for method 'changeColour'
Payload:
[object Object]
Response to method 'changeColour' sent successfully.
aan
```

Direct method

NewAlarm

You can use this tool to invoke direct methods on devices from the cloud. Direct methods have a name, payload, and configurable timeouts. Learn more

Device ID

NewAlarm

Method name *

changeFrequency

Payload

```
{
    "frequency": "500"
}
```

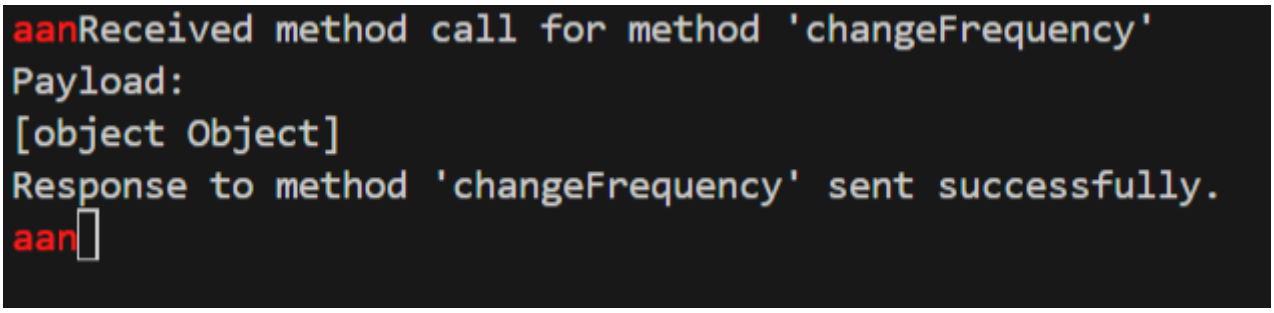
Response timeout: 30 seconds

Connection timeout: Device must already be connected

Invoke method

Result

```
{
    "status": 200,
    "payload": "De frequentie is veranderd naar 500"
}
```



14.

```
PS /home/qing> az iot hub invoke-device-method --hub-name KwiksHub --device-id NewDoor --method-name changeStatus --method-payload '{"status":"open"}'
{
  "payload": "Deur is al open.",
  "status": 200
}
PS /home/qing> az iot hub invoke-device-method --hub-name KwiksHub --device-id NewDoor --method-name changeStatus --method-payload '{"status":"dicht"}'
{
  "payload": "Deur is nu dicht.",
  "status": 200
}
PS /home/qing> az iot hub invoke-device-method --hub-name KwiksHub --device-id NewDoor --method-name changeStatus --method-payload '{"status":"dicht"}'
{
  "payload": "Deur is al dicht.",
  "status": 200
}
PS /home/qing> az iot hub invoke-device-method --hub-name KwiksHub --device-id NewDoor --method-name changeStatus --method-payload '{"status":"open"}'
{
  "payload": "Deur is nu open.",
  "status": 200
}
PS /home/qing>
```

```
PS C:\Users\Kwik\Documents\HBO-ICT\IOT\Cloud Technology & Security> node --env-file=.env .\Door.js
Connected to device. Registering handlers for methods.
Received method call for method 'changeStatus'
Payload:
[object Object]
open
Response to method 'changeStatus' sent successfully.
Received method call for method 'changeStatus'
Payload:
[object Object]
dicht
Response to method 'changeStatus' sent successfully.
Received method call for method 'changeStatus'
Payload:
[object Object]
dicht
Response to method 'changeStatus' sent successfully.
Received method call for method 'changeStatus'
Payload:
[object Object]
open
Response to method 'changeStatus' sent successfully.
```

```
PS /home/qing> az iot hub invoke-device-method --hub-name KwiksHub --device-id NewAlarm --method-name changeColour --method-payload "{'red':'255', 'green':'0'}"
{
  "payload": "Rood is veranderd naar 255. Groen is veranderd naar 0. ",
  "status": 200
}
PS /home/qing>
```

```
PS C:\Users\Kwik\Documents\HBO-ICT\IOT\Cloud Technology & Security> node --env-file=.env .\Alarm.js
Connected to device. Registering handlers for methods.
   Received method call for method 'changeColour'
Payload:
[object Object]
Response to method 'changeColour' sent successfully.
aan
```

16. De naam, de tags, connectiestatus, update tijd, laatste activiteit, authenticatietype, versie en laatste update.

---

17. Het programma simuleert een termometer met een desired temperatuur ingesteld op 20 graden. Het programma stuurt een waarschuwing als het kouder wordt dan de desired temperatuur.

```
function generateMessage(temperature) {
    var data = '';
    if (temperature >= desiredtemperature) {
        data = "Het is " + temperature + " graden. Lekker warm!";
    }
    else {
        data = "Het is " + temperature + " graden. Let op! Te koud!";
    }
  const message = new Message(data);
  return message;
}

function errorHandler(err) {
  console.error(err.message);
}

var desiredtemperature = "";

function connectHandler() {
  console.log('Client connected');
  // Get the temperature every five seconds. If it is below desired temp
  if (!sendInterval) {
    sendInterval = setInterval(() => {
        const temperature = (Math.random() * 30);
        const message = generateMessage(temperature);
        console.log('Sending message: ' + message.getData());
        client.sendEvent(message, printResultFor('send'));
    }, 5000);
  }
}
```

```
const registry = iothub.Registry.fromConnectionString(process.env.IOTHUB_CONNECTION_STRING);
registry.getTwin("NewThermometer", (err, twin) => {
    if (err) {
      console.error('Fout bij het ophalen van de twin:', err);
    } else {
      desiredtemperature = twin.properties.desired.temperature;
    }
  });
```

```
PS C:\Users\Kwik\Documents\HBO-ICT\IOT\Cloud Technology & Security> node --env-file=.env .\Thermometer.js
(node:30836) [DEP0040] DeprecationWarning: The `punycode` module is deprecated. Please use a userland alternative instead.
(Use `node --trace-deprecation ...` to show where the warning was created)
Client connected
Sending message: Het is 23.470892364336184 graden. Lekker warm!
send status: MessageEnqueued
Sending message: Het is 11.826433741400592 graden. Let op! Te koud!
send status: MessageEnqueued
Sending message: Het is 7.058324449648444 graden. Let op! Te koud!
send status: MessageEnqueued
```