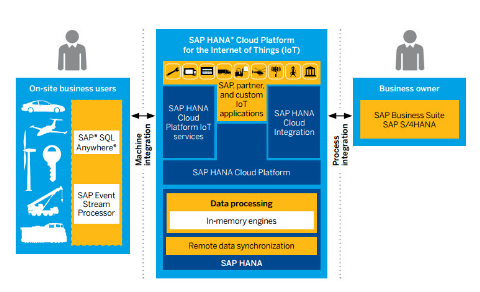
IoT1C08 – Setup a Tessel Device – HCP

|  |  |
| --- | --- |
| **Product**  HANA Cloud Platform IoT Services  **Level**  Undergraduate/Graduate  Beginner  **Focus**  HANA Cloud Platform  **Author** Ross Hightower | MOTIVATION  In this case you will configure a Tessel to feed data into IoT Services.  **PREREQUISITES**  None |



# 

# Setup a Tessel Device

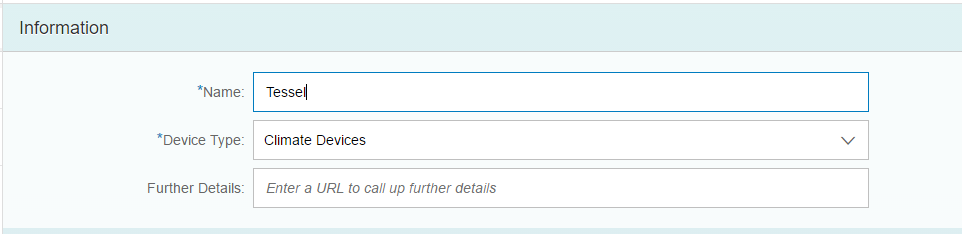
This case uses a Tessel 2 and the instructions for setting up the device can be found here:

<http://tessel.github.io/t2-start/index.html>

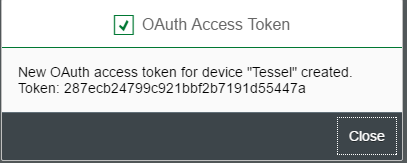
## Configure a Device on HCP IoT Services

Once you have the Tessel climate module functioning you can create a new device on HCP IoT Services Cockpit. You could use the device created previously as long as it uses the Climate Message Type, but it’s a good idea to create a new Device so that you can distinguish where the data comes from.

Log on to the IoT Services Cockpit and click the Devices tile. Create a device called Tessel.



Make sure you save the OAuth token.



## Create the Application

The code shown below was adapted from the Tessel code found at this site:

<https://github.com/saphanaacademy/IoTServices>

The only modifications is the addition of the timestamp field in the message and the interval for sending messages to HCP has been changed to 10 seconds instead of 1 second.

If you ran the test program for the Tessel climate device you can create the new file in the same folder. If you didn’t run the test program yet, go back to the Tessel setup and complete that step.

Create a new file called **climateIoT.js** and copy the code below into it.

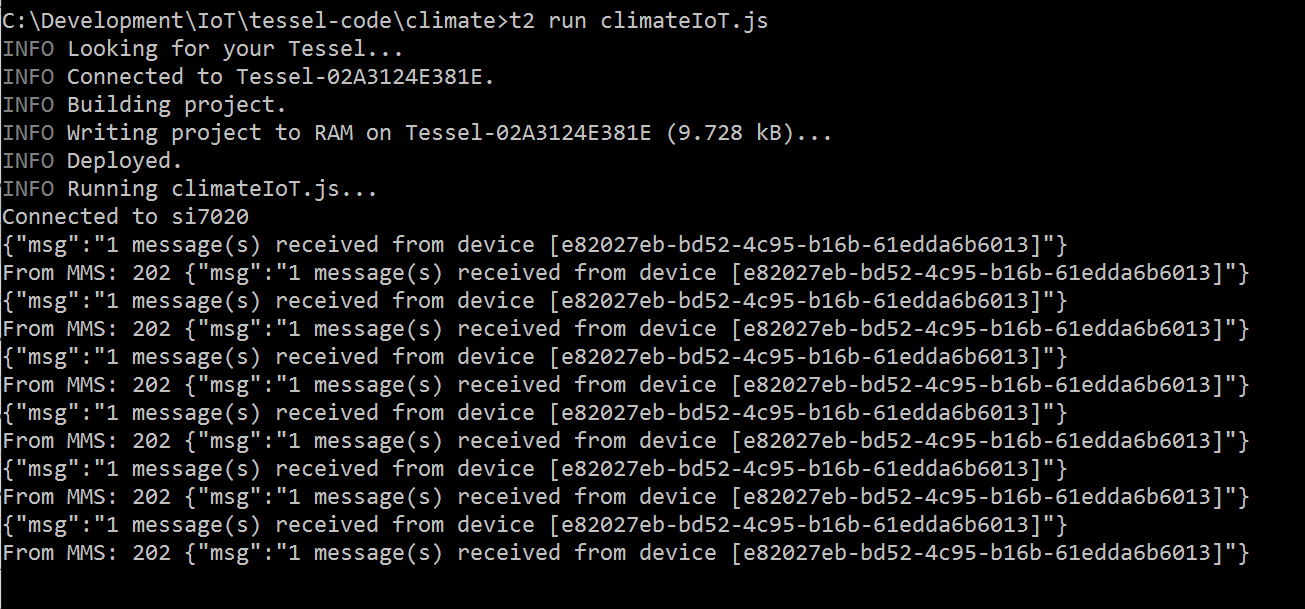
|  |
| --- |
| //Load required modules  var tessel = require('tessel');  var climatelib = require('climate-si7020');  var climate = climatelib.use(tessel.port['A']);  //HCP Connection parameters  var host = 'iotmms<userid>trial.hanatrial.ondemand.com';  var device = '<deviceid>';  var oAuthToken = '<oauthtoken>';  var messageType = '<messagetypeid>';  var path = '/com.sap.iotservices.mms/v1/api/http/data/';  //Sends the message to HCP  function climateHttpSend(temp, humid) {  var http = require('https');  var options = {  host: host,  port: 443,  path: path + device,  agent: false,  headers: {  'Authorization': 'Bearer ' + oAuthToken,  'Content-Type': 'application/json;charset=utf-8'  },  method: 'POST'  };  options.agent = new http.Agent(options);  callback = function(response) {  var body = '';  response.on('data', function (data) {  body += data;  console.log(body);  });  response.on('end', function () {  console.log("From MMS:", response.statusCode, body);  });  response.on('error', function(e) {  console.error(e);  });  }  var req = http.request(options, callback);  req.on('error', function(e) {  console.error(e);  });  req.shouldKeepAlive = true;  var jsonData = {  "mode": "async",  "messageType": messageType,  "messages": [{  "timestamp" : Math.floor(Date.now() / 1000),  "temperature": temp,  "humidity": humid  }]  }  var strData = JSON.stringify(jsonData);  req.write(strData);  req.end();  };  climate.on('ready', function () {  console.log('Connected to si7020');  // Loop forever  setImmediate(function loop () {  climate.readTemperature('f', function (err, temp) {  climate.readHumidity(function (err, humid) {  climateHttpSend(temp, humid);  console.log('Temperature:', temp.toFixed(4) + 'F', 'Humidity:', humid.toFixed(4) + '%RH');  setTimeout(loop, 10000);  });  });  });  });  climate.on('error', function(err) {  console.log('error connecting module', err);  }); |

Listing 1

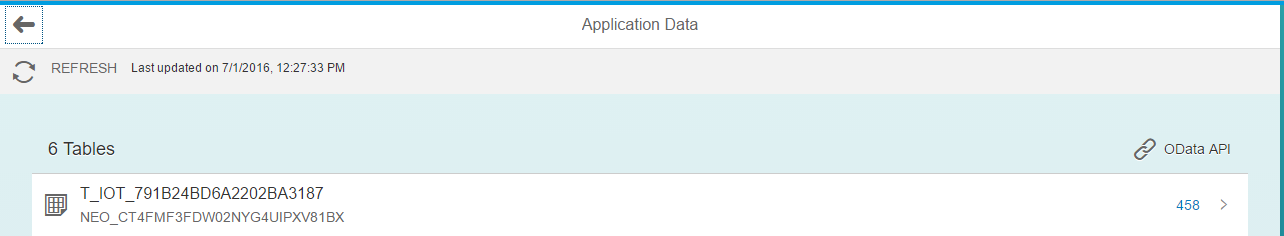
Fill in your HCP parameters at the highlighted parts of the code near the top. Save the file then run it on the Tessel from a command prompt using the following command:

**t2 run climateIoT.js**

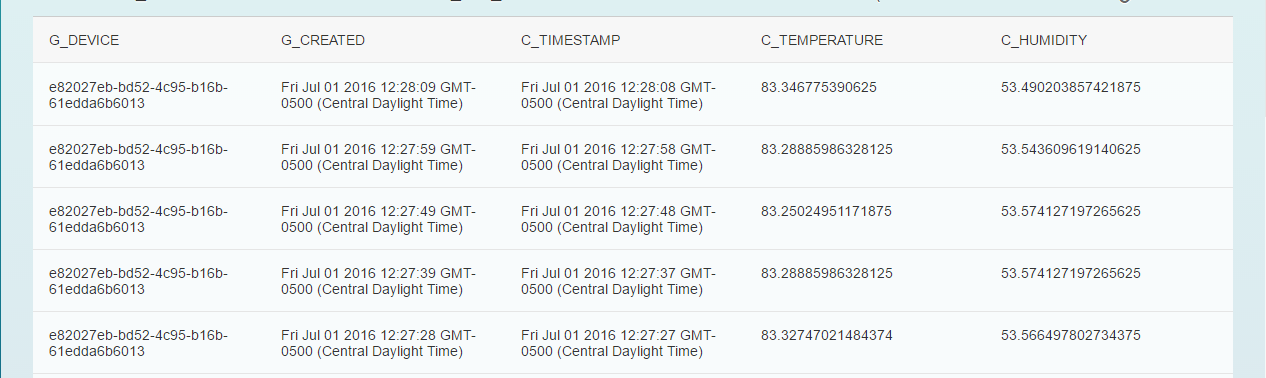
The Tessel will begin sending messages every 10 seconds.



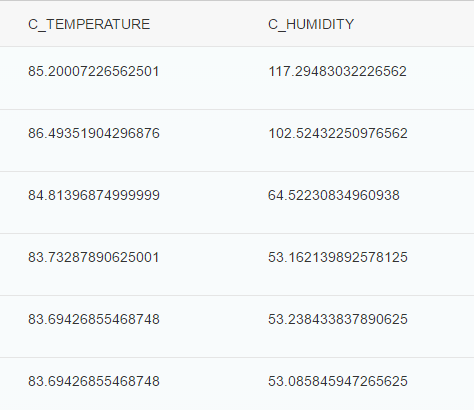
Open the MMS Cockpit and click on the Display stored messages tile then click on the table named for the Climate Message Type.



The newest messages appear at the top of the table:



Cup the climate module in your hand and break gently on it to see the temperature and the humidity rise.



Use ctrl-C at the command prompt to stop sending messages.