AERIS COMMUNICATIONS INC.

Internet of Things Workshop

Sample IoT Project

Aeris is a pioneer and leader in the market of the Internet of Things -- as an operator of end-to-end M2M services and as a technology provider enabling other operators to deliver profitable M2M services. Through our "Made for Machines" technology and services, we strive to fundamentally improve their businesses -- by dramatically reducing costs, improving operational efficiency, reducing time-to-market, and enabling new revenue streams.

THE PROJECT

The purpose of this project is to get introduced to the Internet of Things in practice, including the tools (hardware and software) used to build applications and create solutions to real-world problems. In this Project we will monitor and visualize Real Time temperature and Humidity data using Tessel and Aercloud (an Aeris propriety cloud enablement application platform).

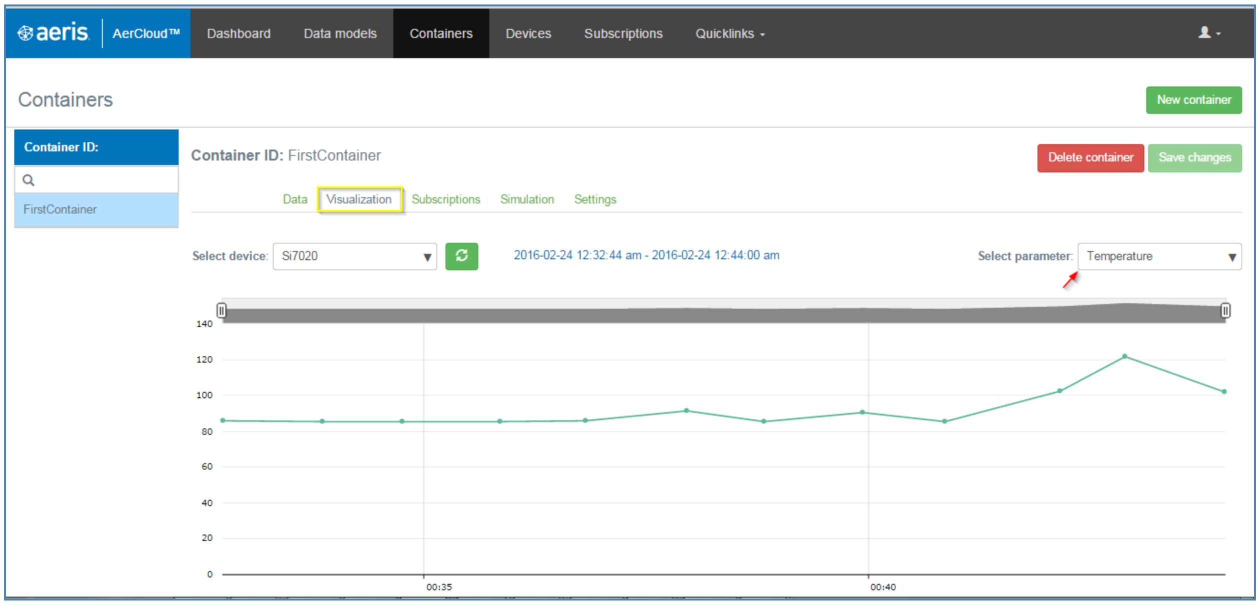


Fig: Real-time temperature reading from Tessel sensor seen on Aercloud application

TASKS:

The project is broadly categorized into four major TASKS as mentioned below. The objective of this project is to build a base Iot applciation which can be further enhanced to build your own "Iot" application.

TASK	TITLE	PAGE
1	VERIFY YOUR AERCLOUD ACCOUNT	4
2	CREATE CONTAINER , DATAMODEL, DEVICE AND SUBSCRIPTION ON ACCOUNT.	6
3	TESSEL BOARD SET UP	10
4	READ TEMPERATURE FROM CLIMATE MODULE AND SEND TO AERLCOUD	14
5	PULL DATA FROM AERCLOUD AND WRITE IT INTO A CSV FILE	18

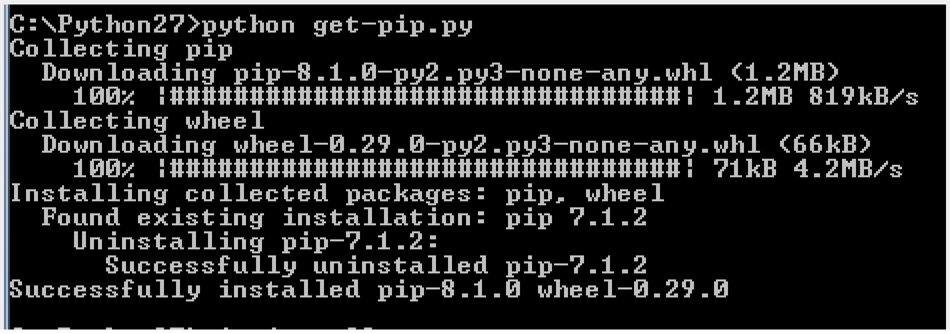
Error! Reference source not found.

PREREQUISITES

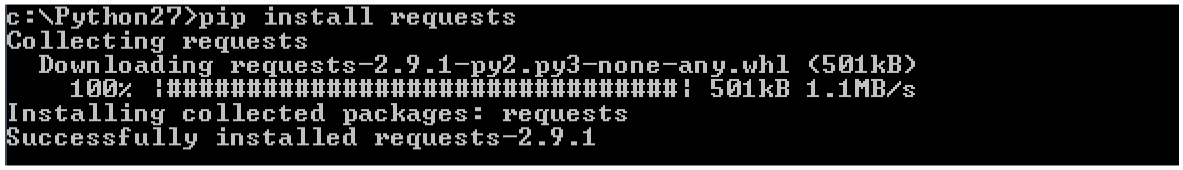
1. **Install python version 2.7.11** from https://www.python.org/downloads/.
Note: You can have more than one versions of python running make sure you download the python package and change the classpath/environment variable to point to the folder where you have downloaded Python2.7.1.

Verification : In terminal/command prompt – Type : `python -V`
Expected Result: `Python 2.7.11` (*or any other version you have installed*)

2. **Install pip**
For Windows:
a)Copy the code from https://bootstrap.pypa.io/get-pip.py into a notepad and save the file as get-pip.py in in C:/Python27/Scripts folder.
b) Then Navigate to in C:/Python27/Scripts folder where the above python script was downloaded and run the command:
```
python get-pip.py
```
For Linux : In the Python installation directory type the following command
```
sudo apt-get install python-pip
```
For Mac: In the Python installation directory type the following command
```
sudo easy_install pip
```

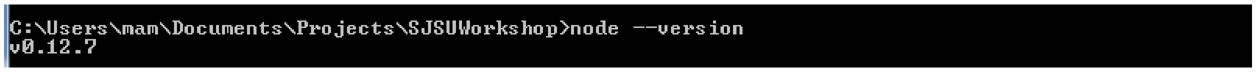

```
C:\Python27>python get-pip.py
Collecting pip
  Downloading pip-8.1.0-py2.py3-none-any.whl (1.2MB)
    100% |################################| 1.2MB 819kB/s
Collecting wheel
  Downloading wheel-0.29.0-py2.py3-none-any.whl (66kB)
    100% |################################| 71kB 4.2MB/s
Installing collected packages: pip, wheel
  Found existing installation: pip 7.1.2
    Uninstalling pip-7.1.2:
      Successfully uninstalled pip-7.1.2
Successfully installed pip-8.1.0 wheel-0.29.0
```

3. Download requests library → In the python installation directory, type the command :
```
pip install requests
```


```
c:\Python27>pip install requests
Collecting requests
  Downloading requests-2.9.1-py2.py3-none-any.whl (501kB)
    100% |################################| 501kB 1.1MB/s
Installing collected packages: requests
Successfully installed requests-2.9.1
```

4. Install Node JS v0.12.7 from https://nodejs.org/en/download/releases/ **PLEASE NOTE** : Tessel1 boards are compatible with Node version **0.12.7**.

Please make sure that the node version downloaded matches 0.12.7.

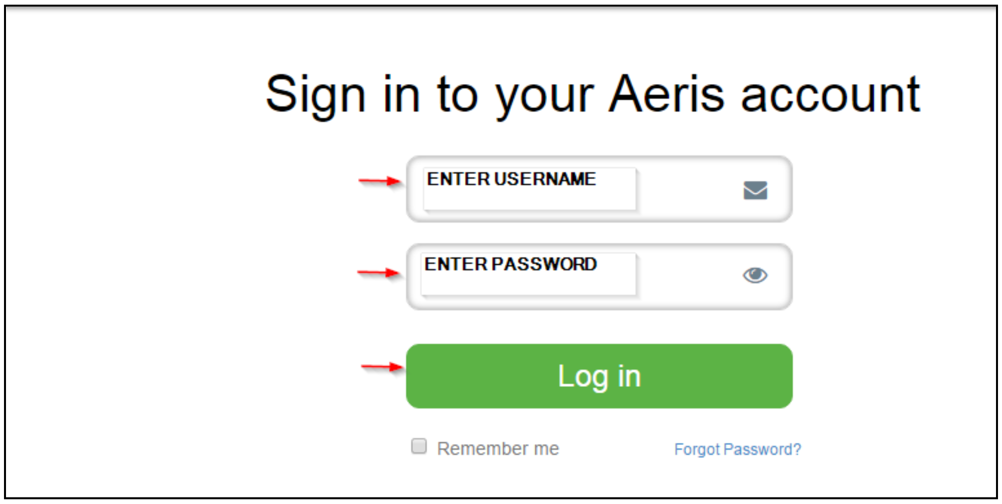

```
C:\Users\mam\Documents\Projects\SJSUWorkshop>node --version
v0.12.7
```

I. VERIFY YOUR AERCLOUD ACCOUNT

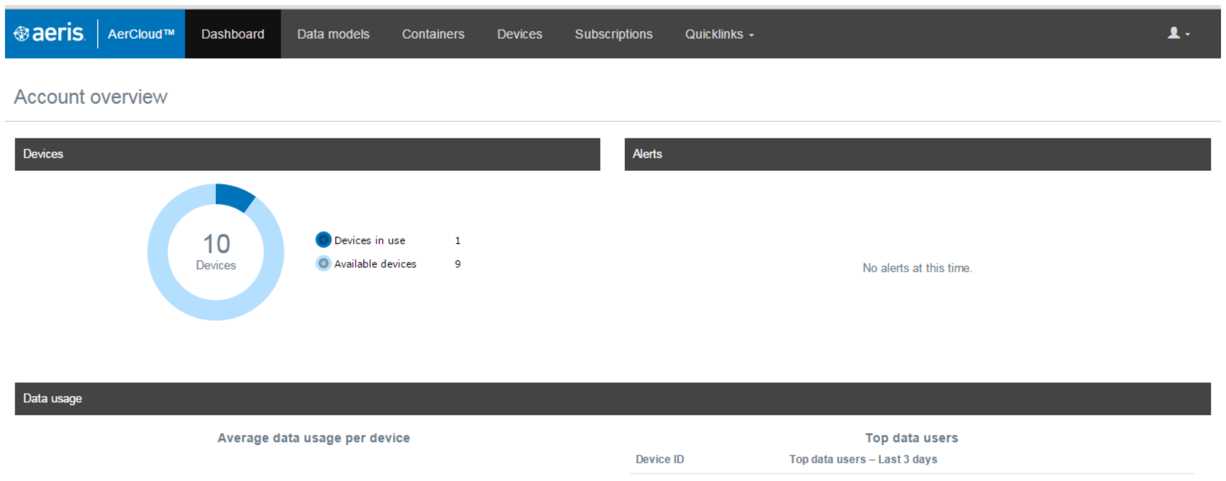
INSTRUCTIONS:

Step1: Collect your username, password ,accountId and other details before you start this sample project.

Click on this link : https://neo.aercloud.aeris.com and sign in with the username and password provided



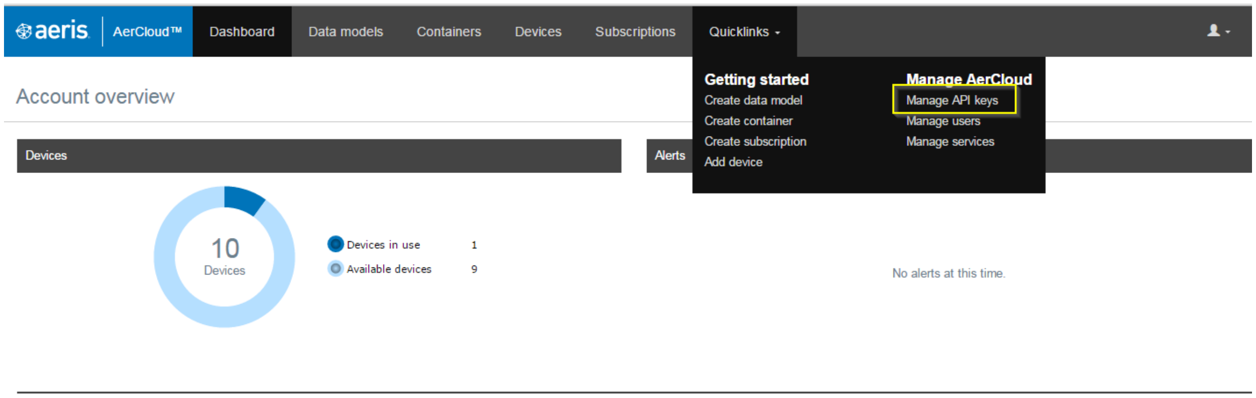
Step -2: You will now be routed to AERCLOUD's dashboard. This is where you will be storing , managing and analyzing the data that your device transmits.



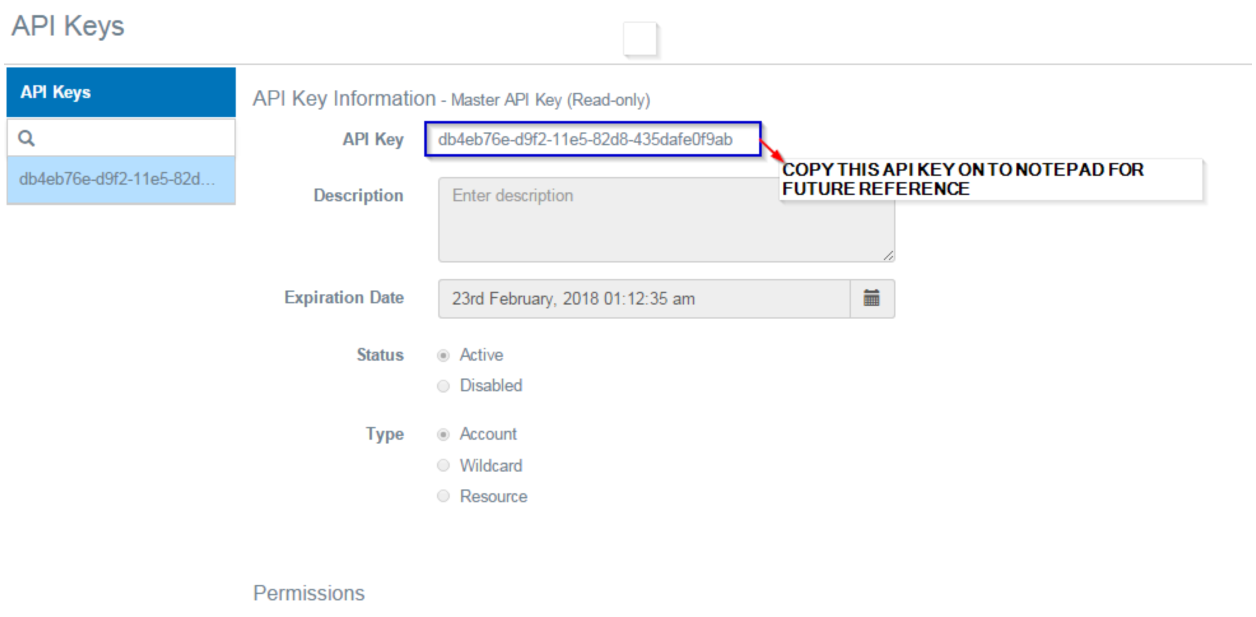
Devices in use	1
Available devices	9

No alerts at this time.

Step -3 : In the Top Navigation bar, Click on “Quicklinks” and select “Manage API Keys” from the dropdown.



Step -4 : You will be taken to the API Keys page. Copy the API Key and paste it on a note pad. You will need this to complete next Task.


API Keys
API Keys
db4eb76e-d9f2-11e5-82d...
API Key Information - Master API Key (Read-only)
COPY THIS API KEY ON TO NOTEPAD FOR FUTURE REFERENCE
Permissions

API Key: db4eb76e-d9f2-11e5-82d8-435dafe0f9ab

Description: Enter description

Expiration Date: 23rd February, 2018 01:12:35 am

Status: [✓ Active] Disabled

Type: [✓ Account] Wildcard Resource

__VALUES TO BE NOTED FROM TASK-1__:

The below two values are needed to proceed to TASK2.

ACCOUNT ID = *This will be provided*

API KEY =

END OF TASK-1

II. CREATE CONTAINER , DATAMODEL, DEVICE AND SUBSCRIPTION ON ACCOUNT.

INSTRUCTIONS

Step -1 : Open note pad and copy the following "Python script" and save the file as "**script.py**" in Python installation folder- in this case : C:/Python27. You can directly use the Script.py in "**TASK-2"** folder in the USB provided to you.

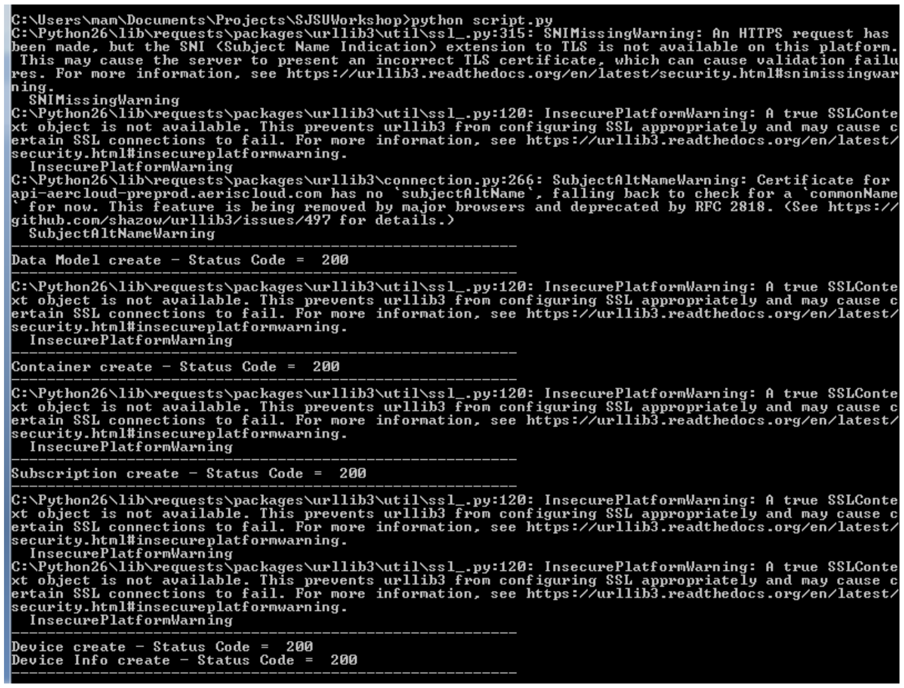
```
import requests

#Create Data Model - id : My First Data Model
url = 'https://api.aercloud.aeris.com/v1/'+'Enter-your-accountId'+'/scls/dataModels'
params = {"apiKey":"Enter-your-api-key"}
data =
'{"id":"FirstDM","sclDataSchema":{"encodings":["JSON","CSV"],"parameters":[{"type":"FL
OAT","name":"Temperature","metainfo":{"uom":"Farenheit"}},{"type":"FLOAT","name":"Humi
dity","metainfo":{"uom":"RH Percentage"}}],"encoding":"JSON"},"name":"First Data
Model","description":"First Data Model for Account"}'
headers = {"Content-type": "application/json"}
try:
 response = requests.post(url, params=params, data=data, headers=headers)
 print "----------------------------------------------------------"
 print "Data Model create - Status Code = ",response.status_code
 print "----------------------------------------------------------"
 if response.status_code == 200:

   #Create Container - id : FirstContainer
   url = 'https://api.aercloud.aeris.com/v1/'+'Enter-your-accountId'+'/containers'
   params = {"apiKey":"Enter-your-acccountApiKey"}
   data = '{"id":"FirstContainer","sclDataModelId":"FirstDM"}'
   headers = {"Content-type": "application/json"}
   response = requests.post(url, params=params, data=data, headers=headers)
   print "----------------------------------------------------------"
   print "Container create - Status Code = ",response.status_code
   print "----------------------------------------------------------"
   if response.status_code == 200:

    #Create Subscription - id : FirstSubs
    url = 'https://api.aercloud.aeris.com/v1/'+'Enter-your-
accountId'+'/containers/subscriptions'
    params = {"apiKey":"Enter-your-acccountApiKey"}
    data =
'{"id":"FirstSubs","subscriptionType":"LONGPOLLING","rule":{"assumptions":[]},"contain
erIds":["FirstContainer"],"contact":"","description":"My First Subscription"}'
    headers = {"Content-type": "application/json"}
    response = requests.post(url, params=params, data=data, headers=headers)
    print "----------------------------------------------------------"
    print "Subscription create - Status Code = ",response.status_code
    print "----------------------------------------------------------"
   else:
      print "ERROR : Container Creation Failed. Please check for valid API Key and
Account number"
 else:
  print "ERROR : Data Model  Creation Failed. Please check for valid API Key and
Account number"
except Exception, e:
 print "EXCEPTION!!-",e
```

```
#Create Device- id : ClimateDevice
url = 'https://api.aercloud.aeris.com/v1/'+'Enter-your-accountId'+'/scls'
params = {"apiKey":"Enter-your-acccountApiKey"}
data = '{"groups":[],"sclId":"Si7020"}'
headers = {"Content-type": "application/json"}
url_info = 'https://api.aercloud.aeris.com/v1/'+'Enter-your-
accountId'+'/scls/Si7020/mgmtObjs/etsiDeviceInfo'
data_info = '{"deviceLabel":"si7020","manufacturer":"Tessel","deviceType":"Climate"}'
try:
 response = requests.post(url, params=params, data=data, headers=headers)
 response_info = requests.post(url_info, params=params, data=data_info,
headers=headers)
 print "----------------------------------------------------------"
 print "Device create - Status Code = ",response.status_code
 print "Device Info create - Status Code = ",response_info.status_code
 print "----------------------------------------------------------"
except Exception, e:
 print "EXCEPTION!!-",e
```

Step – 2: In the above created python script file – "Script.py" , replace values for **Enter-your-accountId** and **<** `your-api-key` **>** which you have obtained **at the end of TASK-1**. Save the file.

Step-3: Navigate to the folder where you saved your Script.py. Then execute the python script using the command – **`python script.py`**

Expected: You should see the response codes = 200 in the terminal-like below.



```
C:\Users\mam\Documents\Projects\SJSUWorkshop>python script.py
C:\Python26\lib\requests\packages\urllib3\util\ssl_.py:315: SNIMissingWarning: An HTTPS request has
been made, but the SNI (Subject Name Indication) extension to TLS is not available on this platform.
 This may cause the server to present an incorrect TLS certificate, which can cause validation failu
res. For more information, see https://urllib3.readthedocs.org/en/latest/security.html#snimissingwar
ning.
  SNIMissingWarning
C:\Python26\lib\requests\packages\urllib3\util\ssl_.py:120: InsecurePlatformWarning: A true SSLConte
xt object is not available. This prevents urllib3 from configuring SSL appropriately and may cause c
ertain SSL connections to fail. For more information, see https://urllib3.readthedocs.org/en/latest/
security.html#insecureplatformwarning.
  InsecurePlatformWarning
C:\Python26\lib\requests\packages\urllib3\connection.py:266: SubjectAltNameWarning: Certificate for
api-aercloud-preprod.aeriscloud.com has no `subjectAltName`, falling back to check for a `commonName
` for now. This feature is being removed by major browsers and deprecated by RFC 2818. (See https://
github.com/shazow/urllib3/issues/497 for details.)
  SubjectAltNameWarning
----------------------------------------------------------
Data Model create - Status Code =  200
----------------------------------------------------------
C:\Python26\lib\requests\packages\urllib3\util\ssl_.py:120: InsecurePlatformWarning: A true SSLConte
xt object is not available. This prevents urllib3 from configuring SSL appropriately and may cause c
ertain SSL connections to fail. For more information, see https://urllib3.readthedocs.org/en/latest/
security.html#insecureplatformwarning.
  InsecurePlatformWarning
----------------------------------------------------------
Container create - Status Code =  200
----------------------------------------------------------
C:\Python26\lib\requests\packages\urllib3\util\ssl_.py:120: InsecurePlatformWarning: A true SSLConte
xt object is not available. This prevents urllib3 from configuring SSL appropriately and may cause c
ertain SSL connections to fail. For more information, see https://urllib3.readthedocs.org/en/latest/
security.html#insecureplatformwarning.
  InsecurePlatformWarning
----------------------------------------------------------
Subscription create - Status Code =  200
----------------------------------------------------------
C:\Python26\lib\requests\packages\urllib3\util\ssl_.py:120: InsecurePlatformWarning: A true SSLConte
xt object is not available. This prevents urllib3 from configuring SSL appropriately and may cause c
ertain SSL connections to fail. For more information, see https://urllib3.readthedocs.org/en/latest/
security.html#insecureplatformwarning.
  InsecurePlatformWarning
C:\Python26\lib\requests\packages\urllib3\util\ssl_.py:120: InsecurePlatformWarning: A true SSLConte
xt object is not available. This prevents urllib3 from configuring SSL appropriately and may cause c
ertain SSL connections to fail. For more information, see https://urllib3.readthedocs.org/en/latest/
security.html#insecureplatformwarning.
  InsecurePlatformWarning
----------------------------------------------------------
Device create - Status Code =  200
Device Info create - Status Code =  200
----------------------------------------------------------
```

STATUS CODES:

When you run the above scripts you might see the following responses and remedies to overcome it

Status code = 200 → Success! Nothing else to do. Verify the configuration in the portal

Status code = 409 → Duplicate . Verify in portal if you that configuration already in the portal

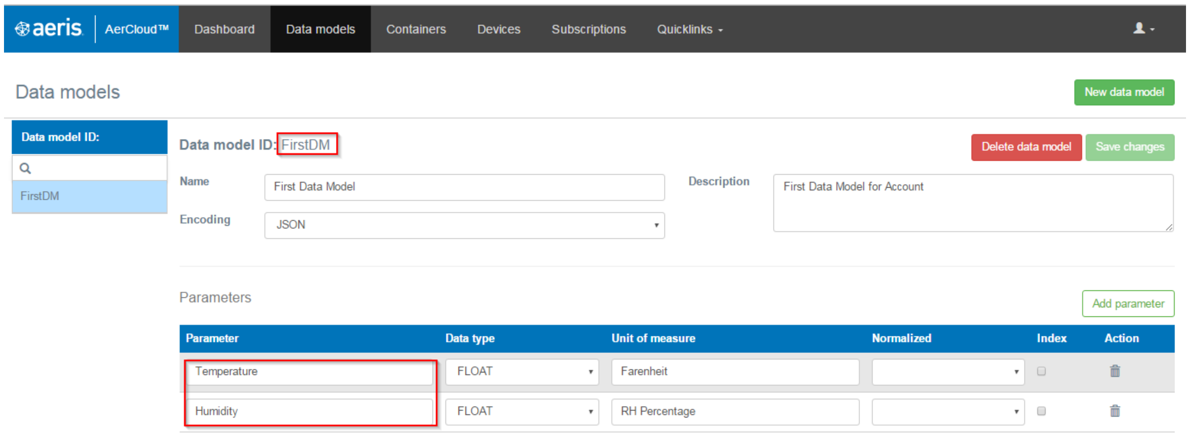
Statuscode=401 →Invalid APIKey. Please re-check if you have entered the APIKey correctly and if you have the correct single quotes around the apikey.

The following have now been created

- One Data model → id: FirstDM . This has two parameters : Temperature and Humidty
- One Container → id: FirstContainer
- One Subscription → id: FirstSubs
- On Device → id: ClimateDevice

Step-4: Now, you can login to Aercloud application and verify that Data Model, Container and Subscriptions are created.

To verify Data Model is created – Click on Data models on the black top navigation bar. You will see FirstDM on the Data models page



Data models

Data model ID: FirstDM

Name: First Data Model

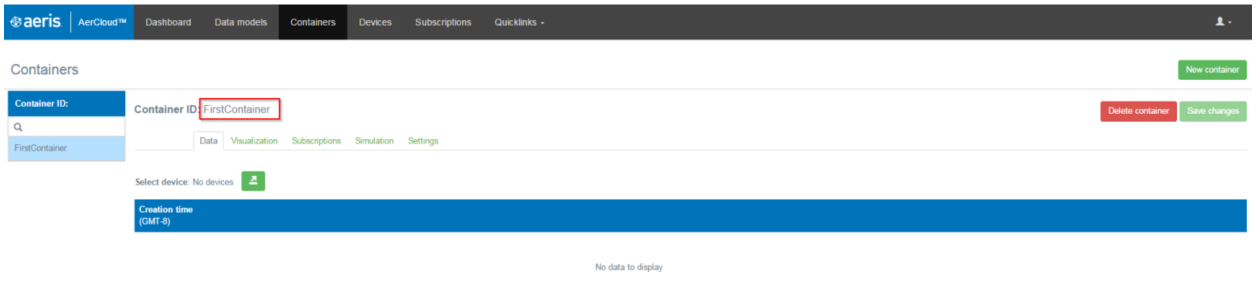
Encoding: JSON

Description: First Data Model for Account

Parameters

Parameter	Data type	Unit of measure	Normalized	Index	Action
Temperature	FLOAT	Farenheit			
Humidity	FLOAT	RH Percentage			

To verify Container is created – Click on **Containers** on the black top navigation bar. You will see FirstContainer on the Containers Page.



Containers

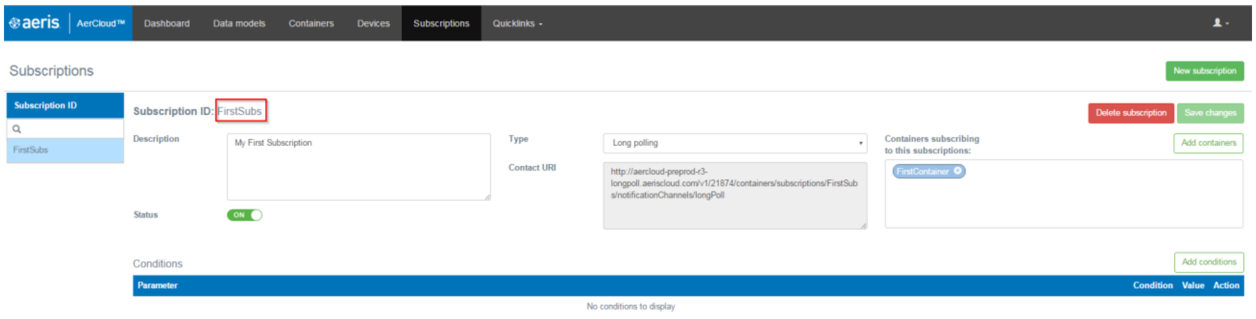
Container ID: FirstContainer

Select device: No devices

Creation time (GMT-8)

No data to display

To verify Subscription is created – Click on **Subscription** on the black top navigation bar. You will see FirstSubs on the Subscriptions Page.



Subscriptions

Subscription ID: FirstSubs

FirstSubs

Description: My First Subscription

Status: ON

Type: Long polling

Contact URI: http://aercloud-preprod-r3-longpoll.aeriscloud.com/v1/21874/containers/subscriptions/FirstSubs/notificationChannels/longPoll

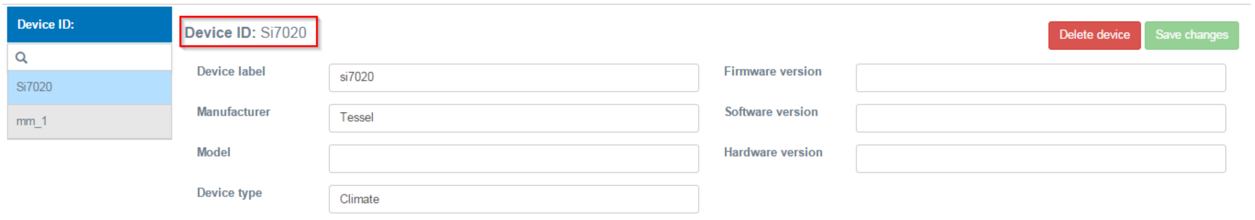
Containers subscribing to this subscriptions: FirstContainer

Conditions

Parameter	Condition	Value	Action

No conditions to display

To verify Device is created – Click on **Devices** on the black top navigation bar. You will see Si7020 created on the Devices Page.



Si7020

mm_1

Device ID: Si7020

Device label	si7020	Firmware version	
Manufacturer	Tessel	Software version	
Model		Hardware version	
Device type	Climate		

END OF TASK – 2

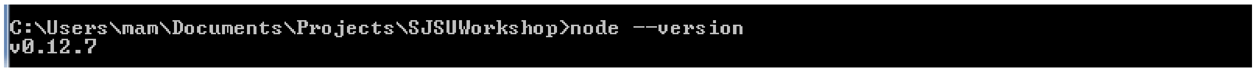
III. TESSEL BOARD SET UP

INSTRUCTIONS

Step-1: Install drivers for Tessel . Usually these drivers are automatically installed.

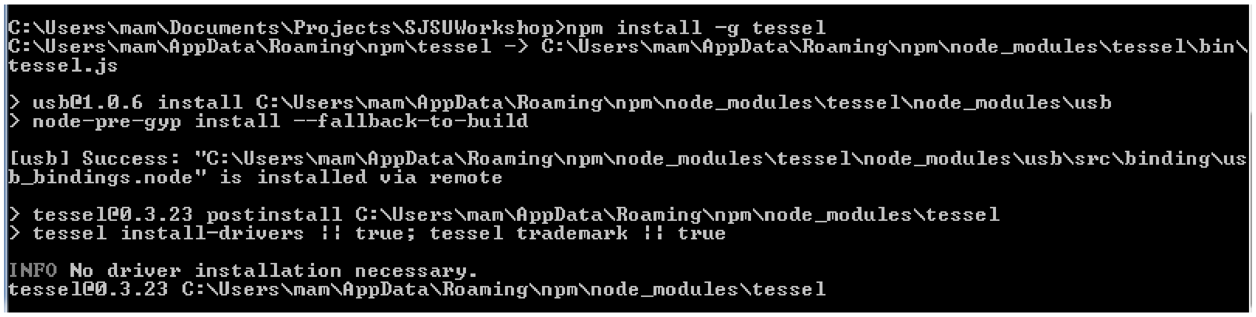
- Note: On windows you might encounter an "Driver not found" issue. In this case, you can go for the option to enable getting drivers from Windows Update which is under "Devices and Printers" -> right click on your computer name -> "Device installation settings".
- If that doesn't work, the manual way to install the driver is to get Zadig208 and bind the Tessel to theWinUSB driver.

Step-2: Please make sure that the node version downloaded matches 0.12.7.



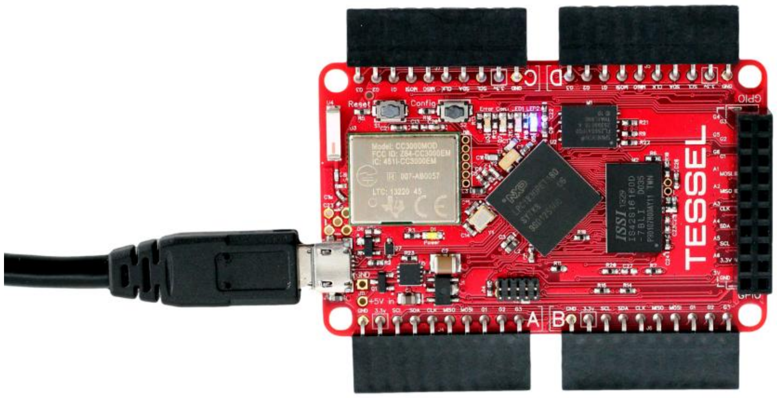
```
C:\Users\mam\Documents\Projects\SJSUWorkshop>node --version
v0.12.7
```

Step-3 : Command : `npm install -g tessel`. If tessel drivers are installed correctly, then you should see something like this in the terminal – in response to the command .

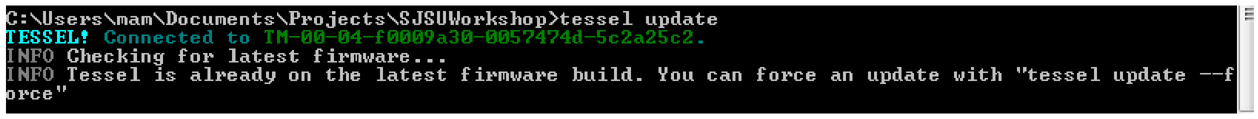


```
C:\Users\mam\Documents\Projects\SJSUWorkshop>npm install -g tessel
C:\Users\mam\AppData\Roaming\npm\tessel -> C:\Users\mam\AppData\Roaming\npm\node_modules\tessel\bin\
tessel.js

> usb@1.0.6 install C:\Users\mam\AppData\Roaming\npm\node_modules\tessel\node_modules\usb
> node-pre-gyp install --fallback-to-build

[usb] Success: "C:\Users\mam\AppData\Roaming\npm\node_modules\tessel\node_modules\usb\src\binding\us
b_bindings.node" is installed via remote

> tessel@0.3.23 postinstall C:\Users\mam\AppData\Roaming\npm\node_modules\tessel
> tessel install-drivers || true; tessel trademark || true

INFO No driver installation necessary.
tessel@0.3.23 C:\Users\mam\AppData\Roaming\npm\node_modules\tessel
```

Step-4 : Connect the Tessel board to your laptop and type the command : `tessel update`.



Most of the boards have been recently updated , so you will see something like this.



```
C:\Users\mam\Documents\Projects\SJSUWorkshop>tessel update
TESSEL! Connected to TM-00-04-f0009a30-0057474d-5c2a25c2.
INFO Checking for latest firmware...
INFO Tessel is already on the latest firmware build. You can force an update with "tessel update --f
orce"
```

Step -5: TEST FOR BLINKING LED LIGHTS (Blinky script)

Type the following commands in Order:

a. mkdir tessel-code
b. cd tessel-code
c. Open notepad . Copy and paste the following and save the file as – package.json in the tessel-code folder

```
{
  "name": "tessel-code",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

d. Open notepad . Copy and paste the following and save the file as – blinky.js in the tessel-code folder

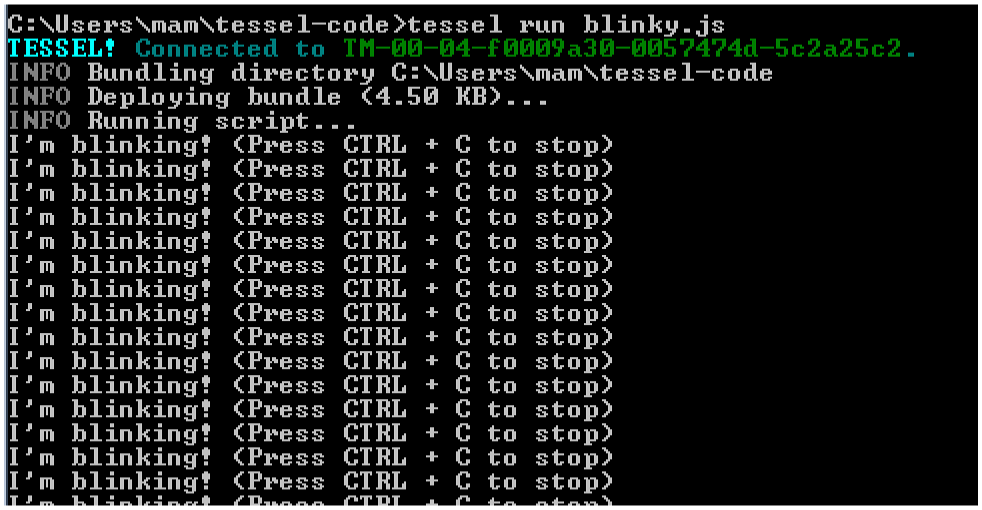
```
// Import the interface to Tessel hardware
var tessel = require('tessel');

// Set the led pins as outputs with initial states
// Truthy initial state sets the pin high
// Falsy sets it low.
var led1 = tessel.led[0].output(1);
var led2 = tessel.led[1].output(0);

setInterval(function () {
    console.log("I'm blinking! (Press CTRL + C to stop)");
    // Toggle the led states
    led1.toggle();
    led2.toggle();
}, 100);
```

e. Navigate to tessel-code folder and type the command s:

```
1. npm init -y
```

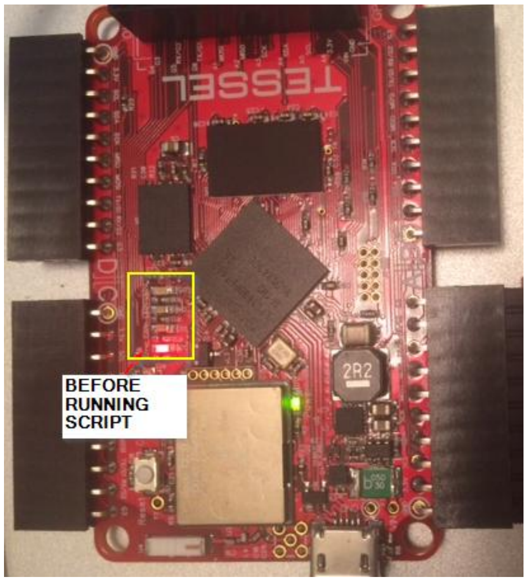
2. `tessel run blinky.js`

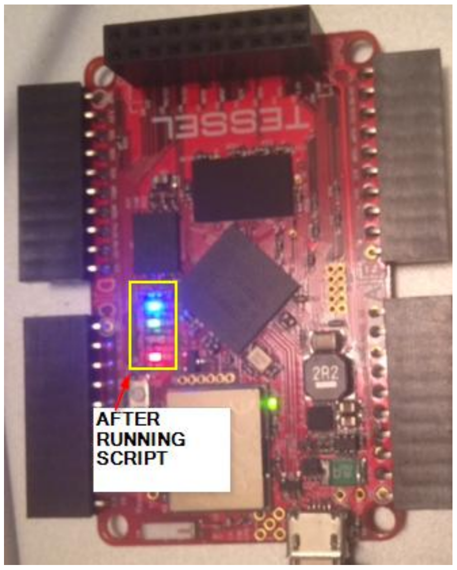
In the terminal you will see the following output. Press Ctrl+C to stop the script.



```
C:\Users\mam\tessel-code>tessel run blinky.js
TESSEL! Connected to TM-00-04-f0009a30-0057474d-5c2a25c2.
INFO Bundling directory C:\Users\mam\tessel-code
INFO Deploying bundle (4.50 KB)...
INFO Running script...
I'm blinking! (Press CTRL + C to stop)
I'm blinking! (Press CTRL + C to stop)
I'm blinking! (Press CTRL + C to stop)
I'm blinking! (Press CTRL + C to stop)
I'm blinking! (Press CTRL + C to stop)
I'm blinking! (Press CTRL + C to stop)
I'm blinking! (Press CTRL + C to stop)
I'm blinking! (Press CTRL + C to stop)
I'm blinking! (Press CTRL + C to stop)
I'm blinking! (Press CTRL + C to stop)
I'm blinking! (Press CTRL + C to stop)
I'm blinking! (Press CTRL + C to stop)
I'm blinking! (Press CTRL + C to stop)
I'm blinking! (Press CTRL + C to stop)
I'm blinking! (Press CTRL + C to stop)
```

On the Tessel board , you can see the lED lights blinking.





Step-6

<u>Internet Credentials have been provided to you</u>

In the terminal , Type the command :

a) `tessel wifi -n <wifi-ssid> -p <password> -t 120`

Note : You can just run tessel wifi –n <***wifi-ssid***> -p <***password***>. The “-t 120” is option . It basically sets the timeout parameter.

In the terminal you will see this:

```
TESSEL! Connected to TM-00-04-f000da30-00624f54-126565c2.
INFO Connecting to "IoT-Workshop" with wpa2 security...
INFO Acquiring IP address.
INFO Connected!
```

This means you are successfully connected to wifi.

Note: If you encounter LIBUSB error here, then please try opening the command prompt with admin access or use Sudo on Linux or Mac.

b) TEST WIFI CONNECTION (OPTIONAL-Run this code AFTER wifi connection is successful.)

At the Tessel-code folder level – create a directory called "wifi" by using the commands

```
1. mkdir wifi
2. cd wifi
3. npm init -y
```

4. Open notepad . Copy and paste the following and save the file as – wifi.js in the wifi folder

```
var http = require('http');
var statusCode = 200;
var count = 1;

setImmediate(function start () {
  console.log('http request #' + (count++))
  http.get("http://httpstat.us/" + statusCode, function (res)
{
   console.log('# statusCode', res.statusCode)
   var bufs = [];
   res.on('data', function (data) {
      bufs.push(new Buffer(data));
      console.log('# received', new Buffer(data).toString());
    })
    res.on('end', function () {
      console.log('done.');
      setImmediate(start);
    })
  }).on('error', function (e) {
    console.log('not ok -', e.message, 'error event')
    setImmediate(start);
  });
});
```

5. Command : `tessel run wifi.js` . Following will be the output in the terminal

```
TESSEL! Connected to TM-00-04-f000da30-00624f54-126565c2.
INFO Bundling directory /Users/maanasamadiraju/tessel-code/wifi
INFO Deploying bundle (5.50 KB)...
INFO Running script...
http request #1
# statusCode 200
# received 200 OK
done.
```

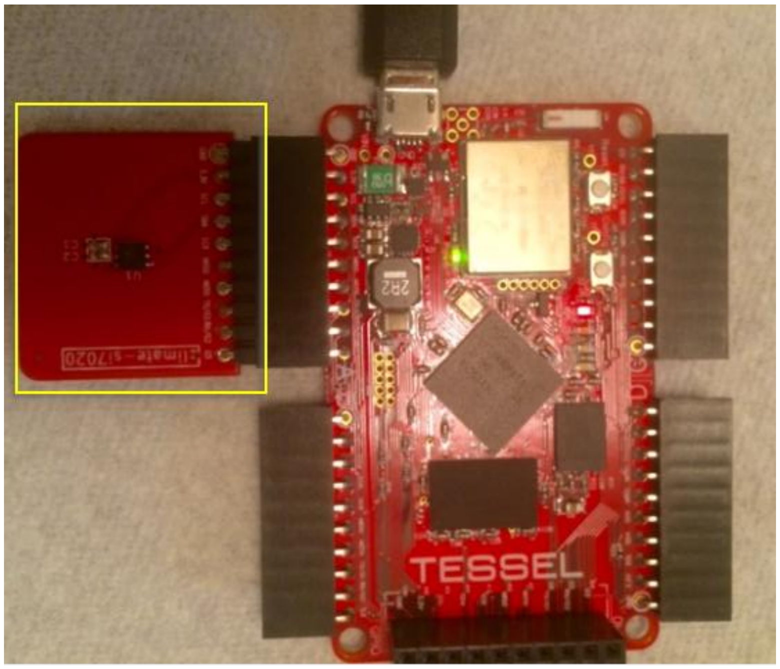
Note: The output from this script will run infinetely until you press **Ctrl+C to terminate the session.**

END OF TASK -3

IV. READ TEMPERATURE FROM CLIMATE MODULE AND SEND TO AERLCOUD

INSTRUCTIONS:

Step-1: Connect Climate module to **PORT – A** of the tessel board which is connected to computer and has wifi set up from Task -3.



Step -2: Type command : `npm install climate-si7020.` This will install **climate-si7020** folder under ..**/node_modules** directory.

Step -3 : Navigate to \climate-si7020\examples folder.You will find "climate.js" script. Erase the old script and copy and paste the below code and Save the climate.js file or use the climate.js file in **TASK-4\climate-si7020\examples** folder in the USB that has been provided to you.

Important: *Plug in your account number and apiKey from Task#1, in place of Enter-your-accountId and Enter-your-acccountApiKey in the code.*

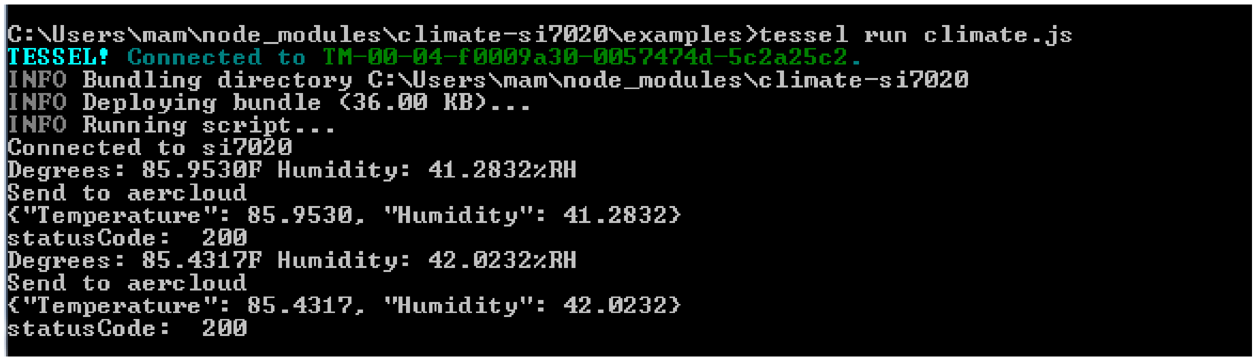
```
// Any copyright is dedicated to the Public Domain.
// http://creativecommons.org/publicdomain/zero/1.0/

/*********************************************
This basic climate example logs a stream
of temperature and humidity to the console.
```

```
*********************************************/

var tessel = require('tessel');
var climatelib = require('../');
var climate = climatelib.use(tessel.port['A']);
var https = require('https');

climate.on('ready', function(){
  console.log("Connected to si7020");
  setImmediate(function loop() {
        climate.readTemperature('f', function(err, temp) {
            climate.readHumidity(function(err, humid) {
                console.log('Degrees:', temp.toFixed(4) + 'F', 'Humidity:',
humid.toFixed(4) + '%RH');
                sendToAercloud(temp.toFixed(4), humid.toFixed(4));
                setTimeout(loop, 60000);
            });
        });
    });
});

climate.on('error', function(err) {
  console.log('error connecting module', err);
});

function sendToAercloud(temp, humid) {
      console.log("Send to aercloud");
    var req = https.request({
        port: 443,
        method: 'POST',
        hostname: 'api.aercloud.aeris.com',
        path: '/v1/Enter-your-
accountId/scls/Si7020/containers/FirstContainer/contentInstances?apiKey='+
'<your-api-key>',
        headers: {
            Host: 'api.aercloud.aeris.com',
            'Accept': 'application/json, text/plain, */*',
                  'Content-Type': 'application/json',
                  'User-Agent': 'tessel'
             }
    }, function(res) {
        console.log('statusCode: ', res.statusCode);
    });
    console.log('{"Temperature": ' + temp + ', "Humidity": ' + humid + '}');
    req.write('{"Temperature": ' + temp + ', "Humidity": ' + humid + '}');
    req.end();
    req.on('error', function(e) {
        console.error("error posting data to your container",e);
    });
}
```

Explaination: The above code reads the temperature and humidity values given by climate module and POSTs this data to Aercloud container created in Task-2.

Step -4 : Run the above script by using the command : `tessel run climate.js`

In the **terminal** , you will see the values being read and being sent to aercloud every 1 minute interval.

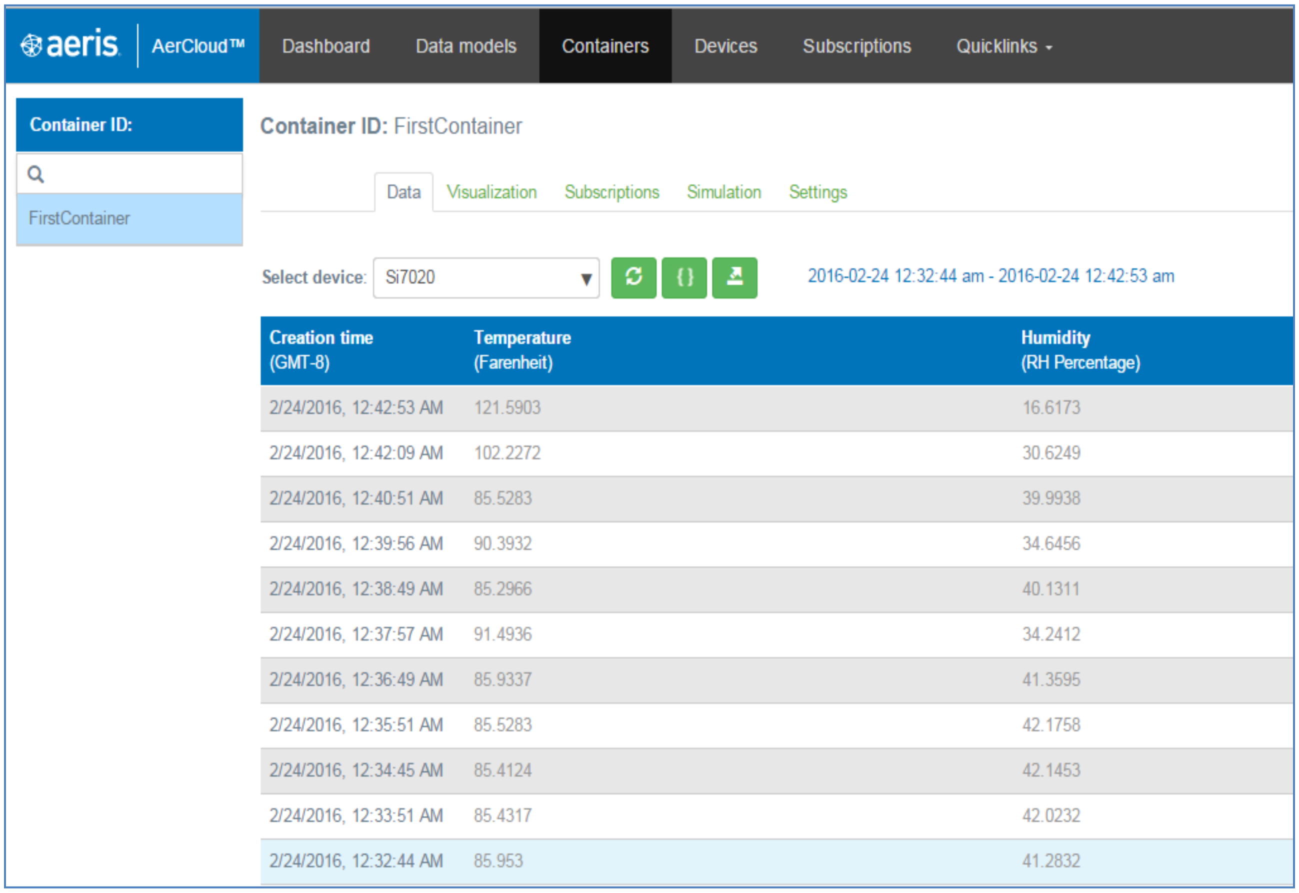


```
C:\Users\mam\node_modules\climate-si7020\examples>tessel run climate.js
TESSEL! Connected to TM-00-04-f0009a30-0057474d-5c2a25c2.
INFO Bundling directory C:\Users\mam\node_modules\climate-si7020
INFO Deploying bundle (36.00 KB)...
INFO Running script...
Connected to si7020
Degrees: 85.9530F Humidity: 41.2832%RH
Send to aercloud
{"Temperature": 85.9530, "Humidity": 41.2832}
statusCode:  200
Degrees: 85.4317F Humidity: 42.0232%RH
Send to aercloud
{"Temperature": 85.4317, "Humidity": 42.0232}
statusCode:  200
```

Step -5: View Data on Aercloud:

Login into Aercloud UI → Click on Container tab from the top black bar and select device = ClimateDevice.

- In the **Data tab** , you can see that the data from the climate module is being published and stored in Aercloud Container.



Container ID: FirstContainer

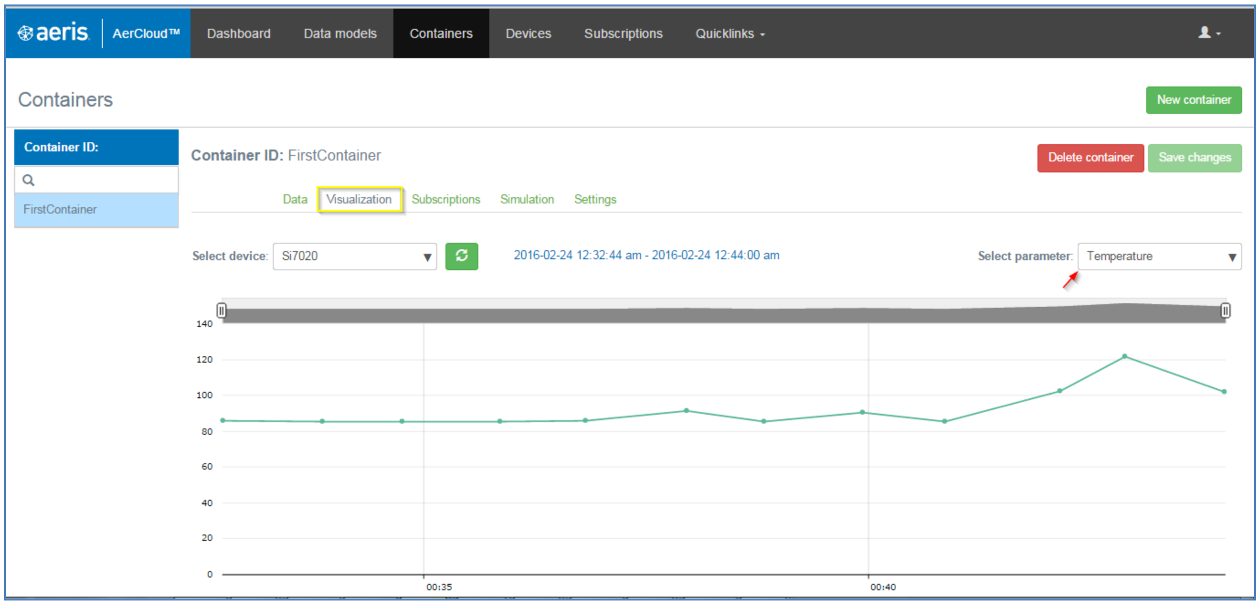
Select device: Si7020

2016-02-24 12:32:44 am - 2016-02-24 12:42:53 am

Creation time (GMT-8)	Temperature (Farenheit)	Humidity (RH Percentage)
2/24/2016, 12:42:53 AM	121.5903	16.6173
2/24/2016, 12:42:09 AM	102.2272	30.6249
2/24/2016, 12:40:51 AM	85.5283	39.9938
2/24/2016, 12:39:56 AM	90.3932	34.6456
2/24/2016, 12:38:49 AM	85.2966	40.1311
2/24/2016, 12:37:57 AM	91.4936	34.2412
2/24/2016, 12:36:49 AM	85.9337	41.3595
2/24/2016, 12:35:51 AM	85.5283	42.1758
2/24/2016, 12:34:45 AM	85.4124	42.1453
2/24/2016, 12:33:51 AM	85.4317	42.0232
2/24/2016, 12:32:44 AM	85.953	41.2832

- In the Visualization tab, you can see the “trend”/graphical representation on how Temperature and Humidity is varying

GRAPHICAL REPRESENTATION OF TEMPERATURE VARIATION:



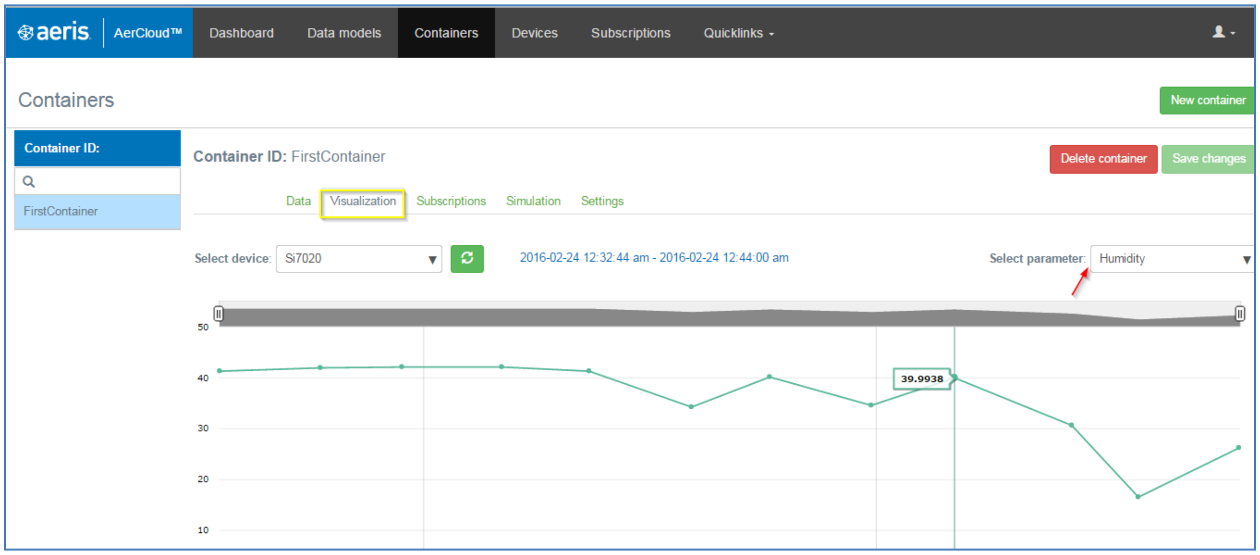
Container ID: FirstContainer

Select device: Si7020

2016-02-24 12:32:44 am - 2016-02-24 12:44:00 am

Select parameter: Temperature

GRAPHICAL REPRESENTATION OF HUMIDITY VARIATION



Container ID: FirstContainer

Select device: Si7020

2016-02-24 12:32:44 am - 2016-02-24 12:44:00 am

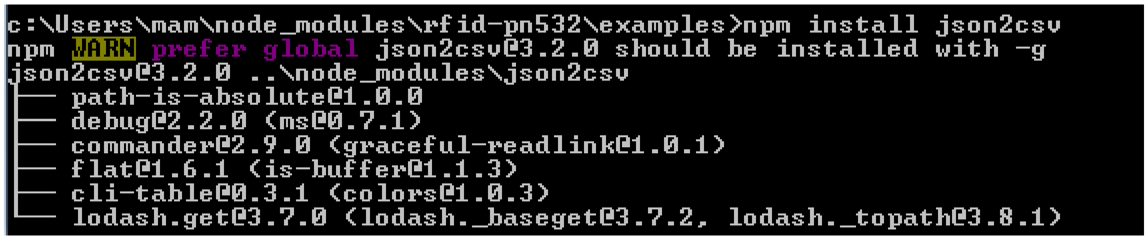
Select parameter: Humidity

END OF TASK -4

V. PULL DATA FROM AERCLOUD AND WRITE IT INTO A CSV FILE

The objective of this task is to be able to access Data from Aercloud and write to an external location. In this sample project, we will be access the Data from Aercloud and write to a CSV file. The code can be modified to write this data to MySQL database or any other such repositories depending on use-cases.

Step-1a: Navigate to /climate-Si7020/examples folder. **Install json2csv library using the command:**
`npm install json2csv`



```
c:\Users\mam\node_modules\rfid-pn532\examples>npm install json2csv
npm WARN prefer global json2csv@3.2.0 should be installed with -g
json2csv@3.2.0 ..\node_modules\json2csv
├── path-is-absolute@1.0.0
├── debug@2.2.0 (ms@0.7.1)
├── commander@2.9.0 (graceful-readlink@1.0.1)
├── flat@1.6.1 (is-buffer@1.1.3)
├── cli-table@0.3.1 (colors@1.0.3)
└── lodash.get@3.7.0 (lodash._baseget@3.7.2, lodash._topath@3.8.1)
```

Step-1b: Open a notepad and copy-paste the below code and save it as – "getClimateDataFromAercloud.js" in /climate-Si7020/examples folder. Please note to enter you accountId and apiKey. This JS file is available in **TASK-5/climate-Si7020/examples** folder in the usb provided to you.

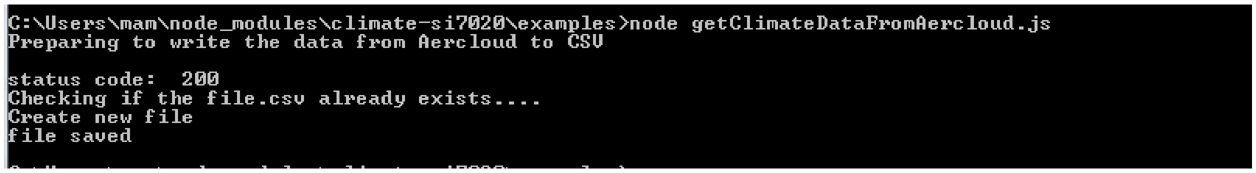
Note: This script would work if the json2 csv is installed at the /climate-Si7020/examples folder level

```
var https = require('https');
var json2csv = require('json2csv');
var fs = require('fs');
var pathFile = 'main/';
var dataResponse='';
var httpResponse='';

writeAercloudDataToCsv(); //function call

function writeAercloudDataToCsv() {
       console.log("Preparing to write the data from Aercloud to CSV");
    /*
    * HTTP Options
    * The below GET call GETs the most recent 100 rows. To get more, add queryparam
    * Eg: url?apiKey=<apiKey>&max =200
    */
    var options = {
        host :  'api.aercloud.aeris.com,
        port : 443,
        path : '/v1/Enter-your-
accountId/scls/Si7020/containers/FirstContainer/contentInstances?apiKey='+ 'Enter-
your-apiKey',
        method : 'GET',
        headers: {
            'Accept': 'application/json, text/plain, */*',
            'Content-Type': 'application/json'
        }
    }
    var getReq = https.request(options, function(res) {
        console.log("\nstatus code: ", res.statusCode); //statuscode = 200 means
success
           //get the Data from the response
           res.on('data', function(data) {
            dataResponse += data;
           });
           //parse the response to JSON
           res.on('end', function() {
            httpResponse = JSON.parse(dataResponse);

           if(!isEmpty(httpResponse)){
                    var contentTypeBinaryData = [];
                    var contentTypeBinaryFields = [];
                    //Prepare the Data array with JSON elements with Temperature,
Humidity and CreateTime data
                    for (var key in httpResponse.contentInstances){
                        var jsonObject =
JSON.parse(httpResponse.contentInstances[key].content.contentTypeBinary);
                        jsonObject["creationTime"] = new
Date(httpResponse.contentInstances[key].creationTime).toLocaleString();
                        contentTypeBinaryData.push(jsonObject);
                    }
                    //Treat each JSON element as key:value pair. Key is the header
                    for (var key in contentTypeBinaryData[0]){
                         contentTypeBinaryFields.push(key);
                    }
                    //Preparing to CSV file.
                    json2csv({ data: contentTypeBinaryData, fields:
contentTypeBinaryFields }, function(err, csv) {
                        if (err) console.log(err);
                        //if we don't specify the path, it takes root of the project ,
                        // e.g. node main/nodeWriteJSONTOCSV , the main path is main/
                        if(!fileExists('file.csv')) {
                             console.log("Create new file");
                            fs.writeFile('file.csv', csv, function(err) {
                            if (err) {
                            console.log("\nERROR:Error writing to a File-Please
verify.",err);
                            } else {
                            console.log('file saved');
                            }
                          });
                        } else {
                            console.log("\nERROR: File already exists. Please rename
the existing file and rerun.");
                        }
                   });
            } else {
                console.log("Empty response received. No Data to write to File.");
            }
        });
    });

 //end the request
    getReq.end();
    getReq.on('error', function(err){
        console.log("Error: ", err);
    });
```

```
//Function: Used to check if the File already exists
function fileExists(filePath)
    {
        console.log("Checking if the file.csv already exists....");
        try
        {
            return fs.statSync(filePath).isFile();
        }
        catch (err)
        {
            return false;
        }
    }

//Funtion: Used to check if the object is empty
var isEmpty = function(obj) {
  return Object.keys(obj).length === 0;}}
```

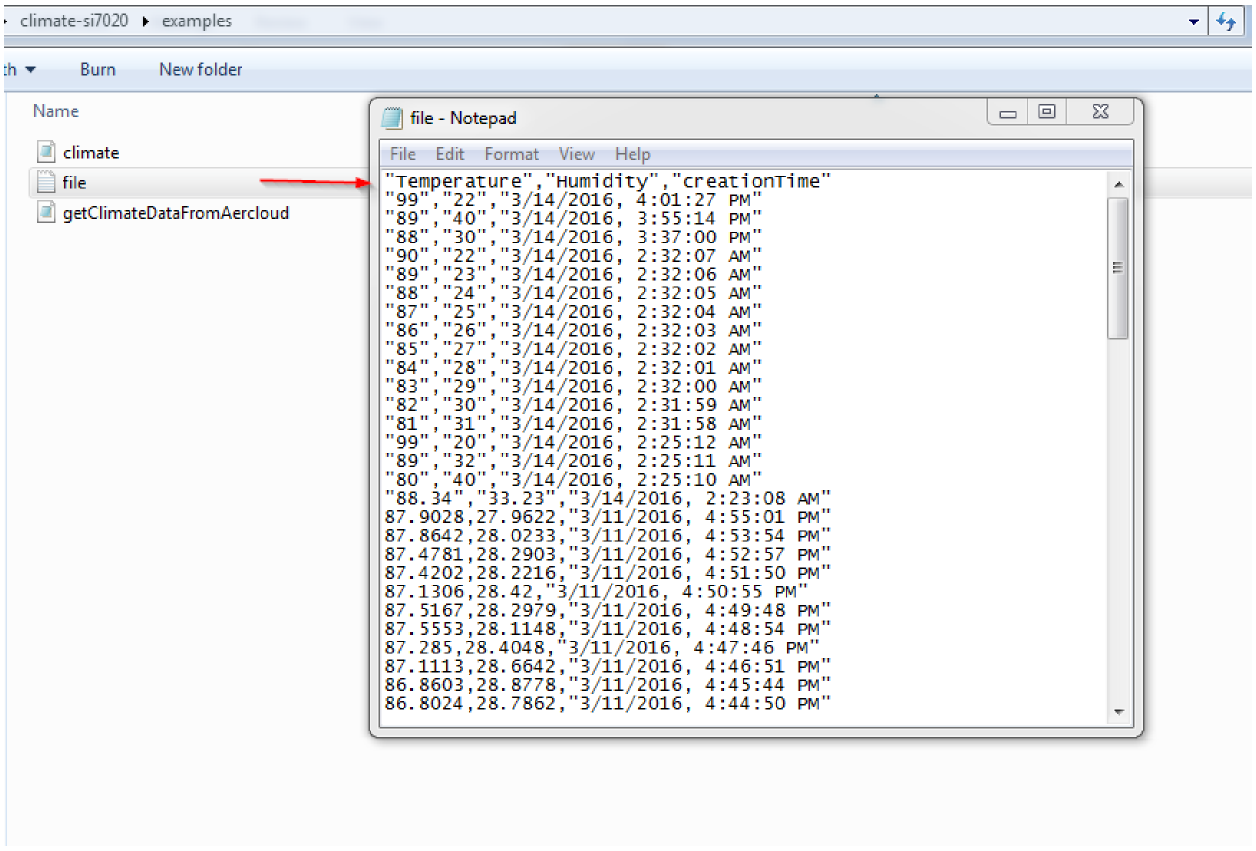
Step-2: Open Command prompt and navigate to /climate-si7020/examples folder (basically the location where you saved the above js file)

Then type the command : `node getClimateDateFromAercloud.js`



```
C:\Users\mam\node_modules\climate-si7020\examples>node getClimateDataFromAercloud.js
Preparing to write the data from Aercloud to CSV

status code:  200
Checking if the file.csv already exists....
Create new file
file saved
```

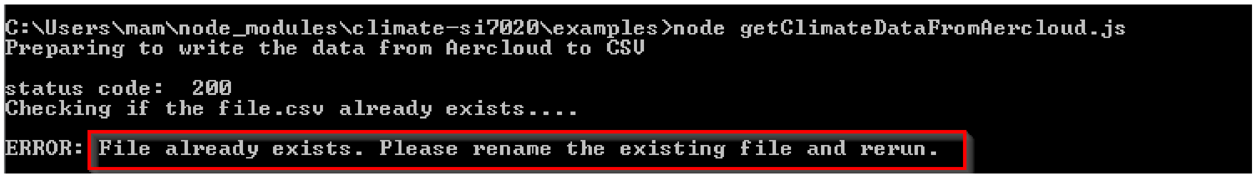
In the same location, you will see that a csv called “file.csv” has been created.


climate-si7020 ▸ examples
Burn
New folder
Name
climate
file
getClimateDataFromAercloud
file - Notepad
File Edit Format View Help

```
"Temperature","Humidity","creationTime"
"99","22","3/14/2016, 4:01:27 PM"
"89","40","3/14/2016, 3:55:14 PM"
"88","30","3/14/2016, 3:37:00 PM"
"90","22","3/14/2016, 2:32:07 AM"
"89","23","3/14/2016, 2:32:06 AM"
"88","24","3/14/2016, 2:32:05 AM"
"87","25","3/14/2016, 2:32:04 AM"
"86","26","3/14/2016, 2:32:03 AM"
"85","27","3/14/2016, 2:32:02 AM"
"84","28","3/14/2016, 2:32:01 AM"
"83","29","3/14/2016, 2:32:00 AM"
"82","30","3/14/2016, 2:31:59 AM"
"81","31","3/14/2016, 2:31:58 AM"
"99","20","3/14/2016, 2:25:12 AM"
"89","32","3/14/2016, 2:25:11 AM"
"80","40","3/14/2016, 2:25:10 AM"
"88.34","33.23","3/14/2016, 2:23:08 AM"
87.9028,27.9622,"3/11/2016, 4:55:01 PM"
87.8642,28.0233,"3/11/2016, 4:53:54 PM"
87.4781,28.2903,"3/11/2016, 4:52:57 PM"
87.4202,28.2216,"3/11/2016, 4:51:50 PM"
87.1306,28.42,"3/11/2016, 4:50:55 PM"
87.5167,28.2979,"3/11/2016, 4:49:48 PM"
87.5553,28.1148,"3/11/2016, 4:48:54 PM"
87.285,28.4048,"3/11/2016, 4:47:46 PM"
87.1113,28.6642,"3/11/2016, 4:46:51 PM"
86.8603,28.8778,"3/11/2016, 4:45:44 PM"
86.8024,28.7862,"3/11/2016, 4:44:50 PM"
```

As you can see , this file contains the entire Data that has been sent to Aercloud.

NOTE: If file.csv already exists in folder, you will get an error saying file already exisits. The resolution for this is to rename the exisiting file.csv to file_1.csv and RERUN the code, you will see updated file.csv created.



```
C:\Users\mam\node_modules\climate-si7020\examples>node getClimateDataFromAercloud.js
Preparing to write the data from Aercloud to CSV

status code:  200
Checking if the file.csv already exists....

ERROR: File already exists. Please rename the existing file and rerun.
```

END OF TASK-5