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 IoT and M2M services and as a technology provider enabling other operators to build profitable IoT businesses. Among our customers are the most demanding users of IoT services today, including Hyundai, Acura, Rand McNally, Leica, and Sprint. Through our technology platform and dedicated IoT and M2M services, we strive to fundamentally improve their businesses — by dramatically reducing costs, improving operational efficiency, reducing time-to-market, and enabling new revenue streams. Visit www.aeris.com or follow us on Twitter @AerisM2M to learn how we can inspire you to create new business models and to participate in the revolution of the Internet of Things.

THE PROJECT

The purpose of this project is to get introduced to the Internet of Things (IoT) 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 connectivity management platform).

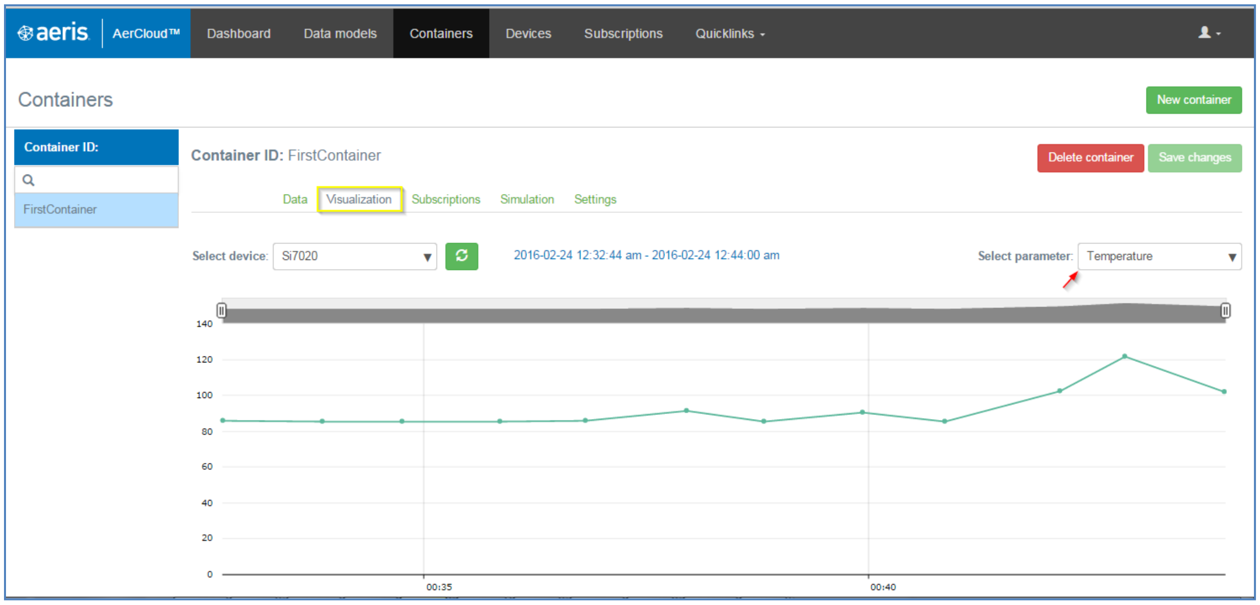


Fig: Real-time temperature reading from Tessel sensor seen on AerCloud application.

TASKS:

The project is broadly categorized into four major tasks outlined below. The objective of this project is to build a base IoT application that can be further enhanced to build your own IoT application.

TASK 1 – SET UP AERCLOUD ACCOUNT

TASK 2 – CREATE CONTAINER, DATA MODEL, DEVICE AND SUBSCRIPTION ON ACCOUNT

TASK 3 – TESSEL BOARD SET UP

TASK 4 – READ TEMPERATURE FROM CLIMATE MODULE AND SEND TO AERCLOUD

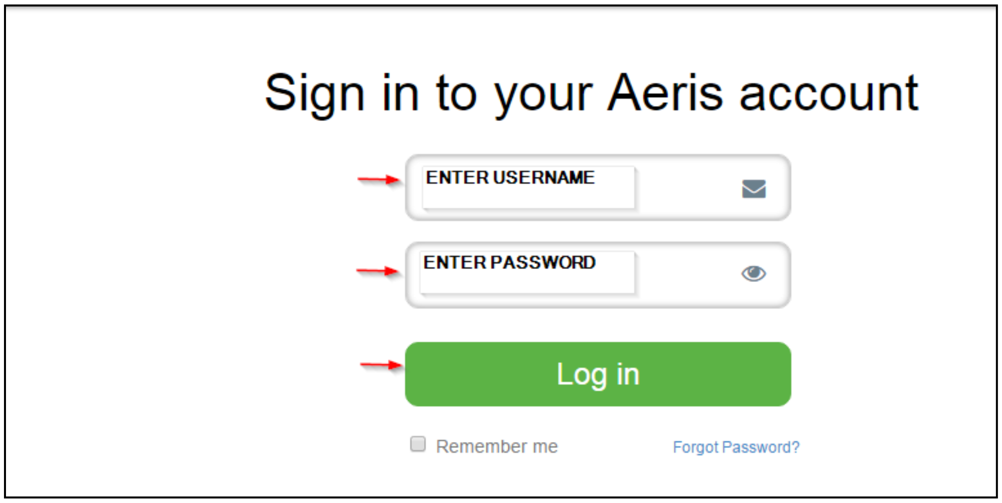
TASK 5 – READ DATA STREAM FROM AERCLOUD TO .CSV FILE

I. SET UP AERCLOUD ACCOUNT

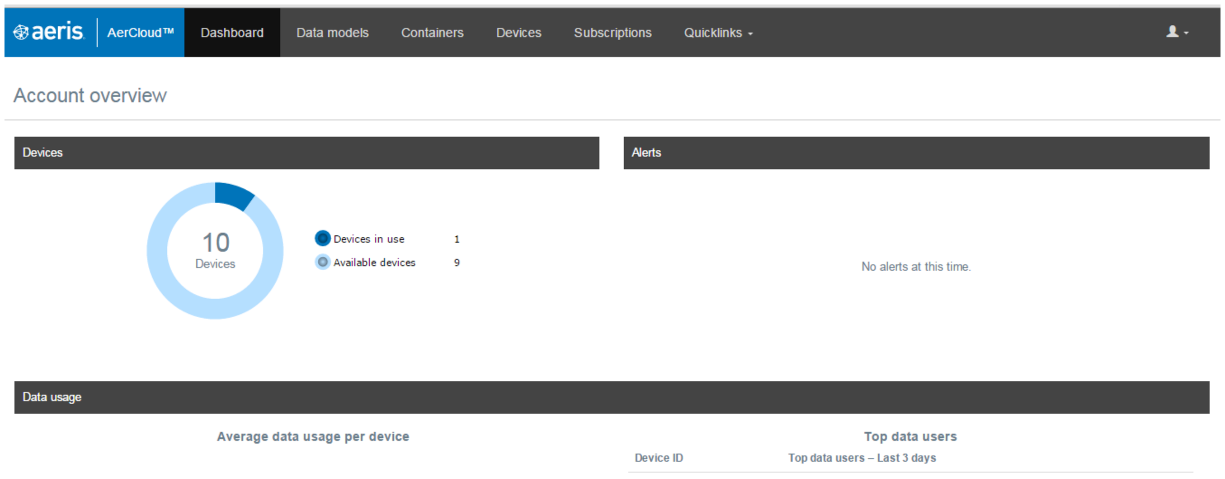
INSTRUCTIONS:

Step 1: Collect your username, password, account ID and other details before you start this sample project.

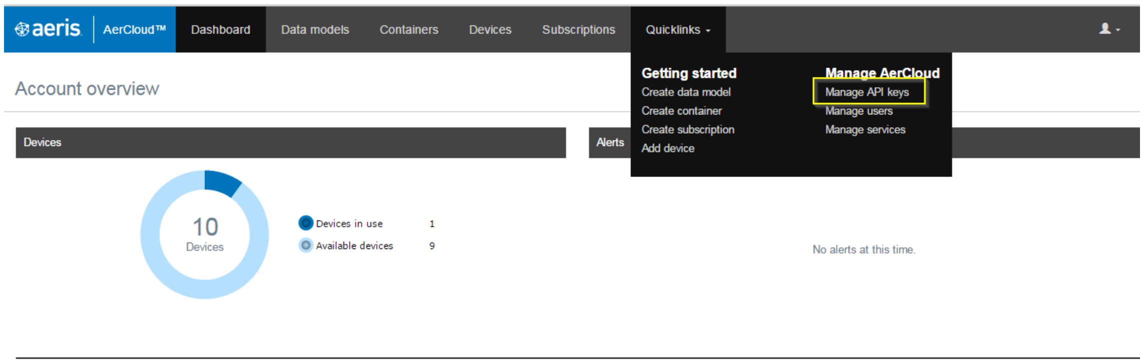
Click on this link: https://neo.aercloud.aeris.com and sign in with the username and password provided.



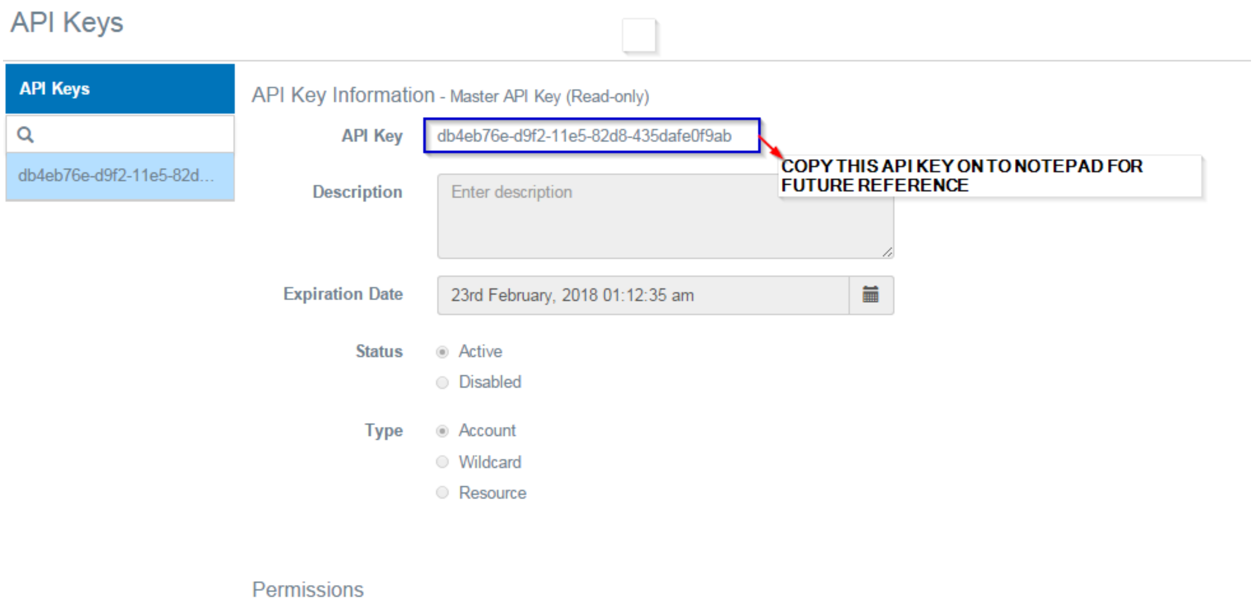
Step 2: You will now be routed to AerCloud's dashboard. This is where you will be storing, managing and analyzing the data that your device transmits.



Step 3: In the top navigation bar, click on "Quicklinks," and select "Manage API Keys" from the drop-down menu.



Step 4: You will be taken to the API Keys page. Copy the API Key, and paste it into Notepad. You will need this to complete next task.



API Key Information - Master API Key (Read-only)

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

COPY THIS API KEY ON TO NOTEPAD FOR FUTURE REFERENCE

Description: Enter description

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

Status: [✓ Active] Disabled

Type: [✓ Account] Wildcard Resource

<u>**VALUES TO BE NOTED FROM TASK 1**</u>:

The below two values are needed to proceed to TASK 2.

ACCOUNT ID = *This will be provided.*
API KEY =

END OF TASK 1

II. CREATE CONTAINER, DATA MODEL, DEVICE AND SUBSCRIPTION ON ACCOUNT

INSTRUCTIONS:

Step 1: Install Python 2.7 (Guide: https://www.python.org/downloads/). Run the installation file and set the environment PATH variable to

Verification: In terminal/command prompt – Type: `python -V`

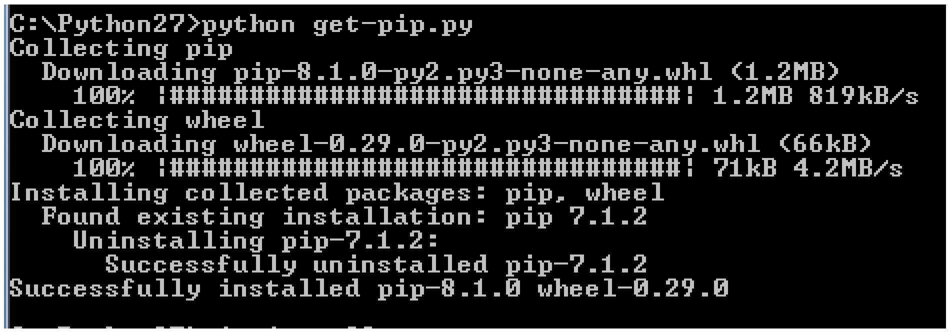
Expected Result: `Python 2.7.11` (*or any other version you have installed*)

Step 2: Download the Requests library. (http://docs.python-requests.org/en/master/user/install/).

Step 2a: Install pip.

For Windows:

1. Copy the code from https://bootstrap.pypa.io/get-pip.py into Notepad, and save the file as get-pip.py in C:/Python27 folder.
2. Then navigate to C:/Python27 folder, and run the command `python get-pip.py`. You will see something like this:



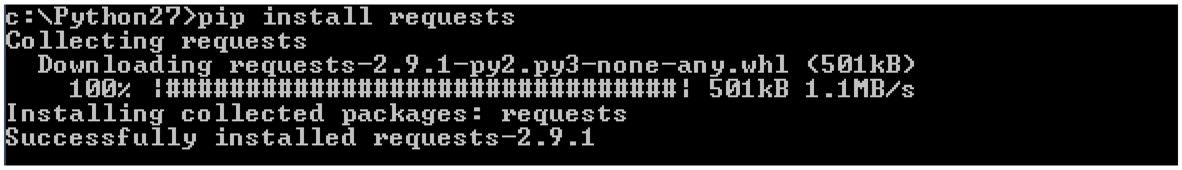
```
C:\Python27>python get-pip.py
Collecting pip
  Downloading pip-8.1.0-py2.py3-none-any.whl (1.2MB)
    100% |################################| 1.2MB 819kB/s
Collecting wheel
  Downloading wheel-0.29.0-py2.py3-none-any.whl (66kB)
    100% |################################| 71kB 4.2MB/s
Installing collected 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
```

For Linux: In the Python installation directory, type the following command:

```
sudo apt-get install python-pip
```

Step 2b: Download the requests library.

In the Python installation directory, type the command – `pip install requests`. You should see something like this:



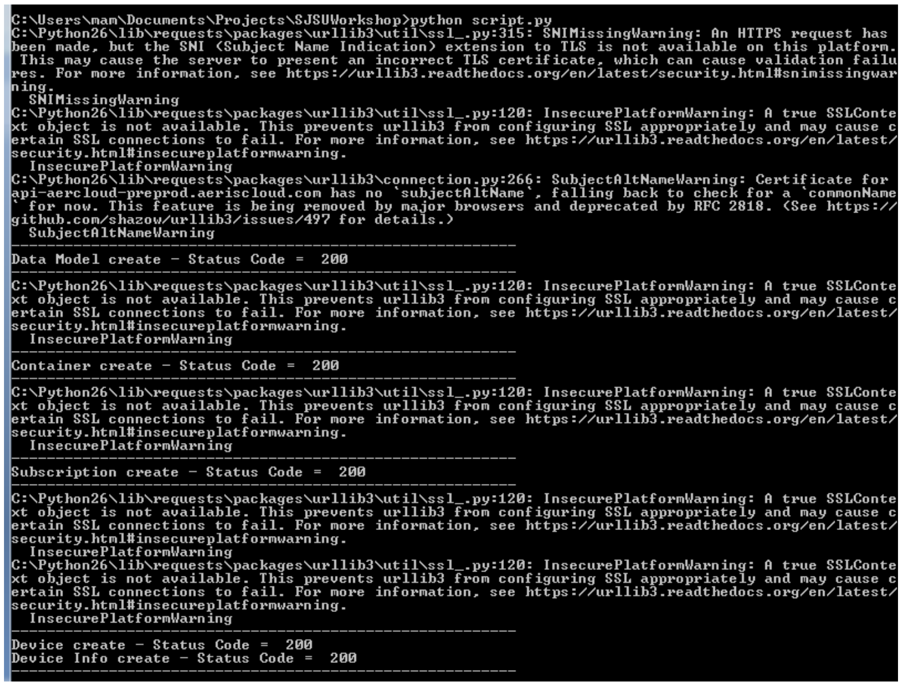
```
c:\Python27>pip install requests
Collecting requests
  Downloading requests-2.9.1-py2.py3-none-any.whl (501kB)
    100% |################################| 501kB 1.1MB/s
Installing collected packages: requests
Successfully installed requests-2.9.1
```

Step 3: Open Notepad. Copy the following Python script, then save the file as “script.py”

```
import requests

#Create Data Model - id : My First Data Model
url = 'https://api-aercloud-
preprod.aeriscloud.com/v1/'+'<accountId>'+'/scls/dataModels'
params = {"apiKey":"<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-preprod.aeriscloud.com/v1/'+'<accountId>'+'/containers'
   params = {"apiKey":"<your-api-key>"}
   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-
preprod.aeriscloud.com/v1/'+'<accountId>'+'/containers/subscriptions'
    params = {"apiKey":"<your-api-key>"}
    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-preprod.aeriscloud.com/v1/'+'<accountId>'+'/scls'
params = {"apiKey":"<your-api-key>"}
data = '{"groups":[],"sclId":"ClimateDevice"}'
headers = {"Content-type": "application/json"}
url_info = 'https://api-aercloud-
preprod.aeriscloud.com/v1/'+'<accountId>'+'/scls/ClimateDevice/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 4: In the above-created Python script file "Script.py" replace values for **<accountID>** and **<apiKey>** that you obtained **at the end of Task 1**. Save the file.

Step 5: 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, as shown below.



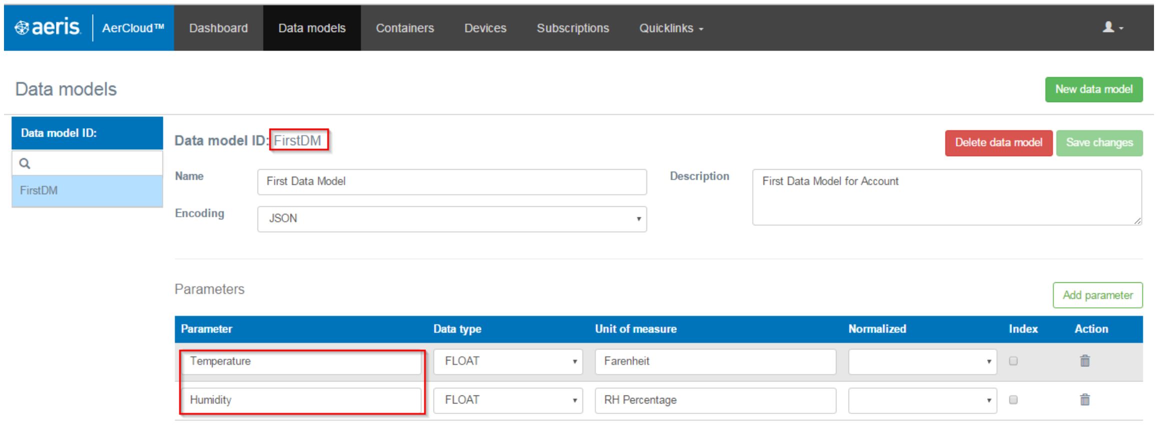
```
C:\Users\mam\Documents\Projects\SJSUWorkshop>python script.py
C:\Python26\lib\requests\packages\urllib3\util\ssl_.py:315: SNIMissingWarning: An HTTPS request has
been made, but the SNI (Subject Name Indication) extension to TLS is not available on this platform.
 This may cause the server to present 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
----------------------------------------------------------
```

The following have now been created:

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

Step 6: Now you can log into the AerCloud application, and verify that the Data Model, Container and Subscriptions are created.

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



Data models

Data model ID:
FirstDM

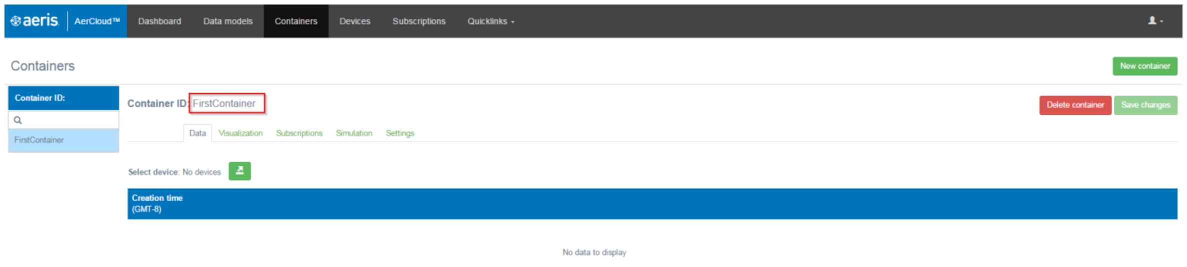
Data model ID: FirstDM

Name	First Data Model	Description	First Data Model for Account
Encoding	JSON		

Parameters

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

To verify the Container is created – Click on “Containers” in the top navigation bar. You will see “FirstContainer” on the Containers page.



Containers

Container ID:
FirstContainer

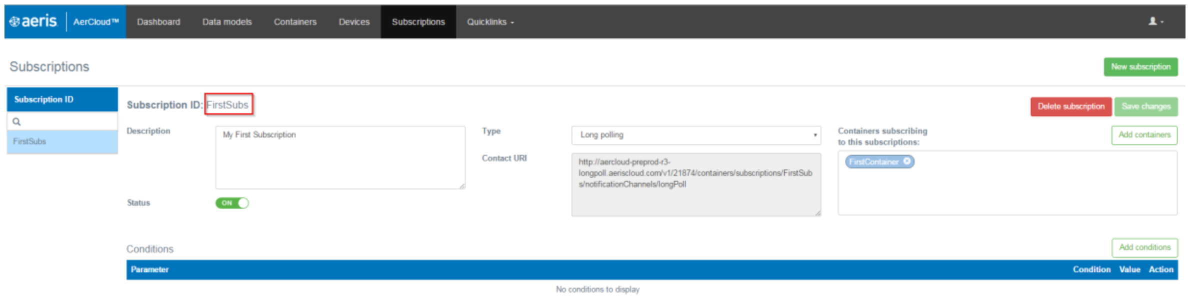
Container ID: FirstContainer

Select device: No devices

Creation time (GMT-8)

No data to display

To verify the Subscription is created – Click on “Subscription” on the top navigation bar. You will see “FirstSubs” on the Subscriptions page.



Subscriptions

Subscription ID
FirstSubs

Subscription ID: FirstSubs

Description	My First Subscription	Type	Long polling
		Contact URI	http://aercloud-preprod-r3-longpoll.aeriscloud.com/v1/21874/containers/subscriptions/FirstSubs/notificationChannels/longPoll
Status	ON		

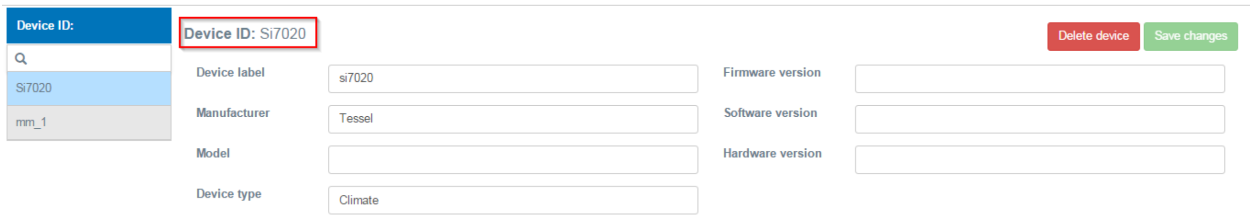
Containers subscribing to this subscriptions: FirstContainer

Conditions

Parameter	Condition	Value	Action

No conditions to display

To verify the Device is created – Click on “Devices” on the top navigation bar. You will see “Si7020” on the Devices page.



Device ID: Si7020

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

END OF TASK 2

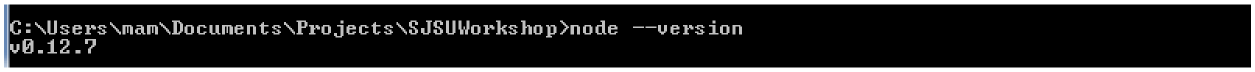
III. TESSEL BOARD SET UP

INSTRUCTIONS:

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

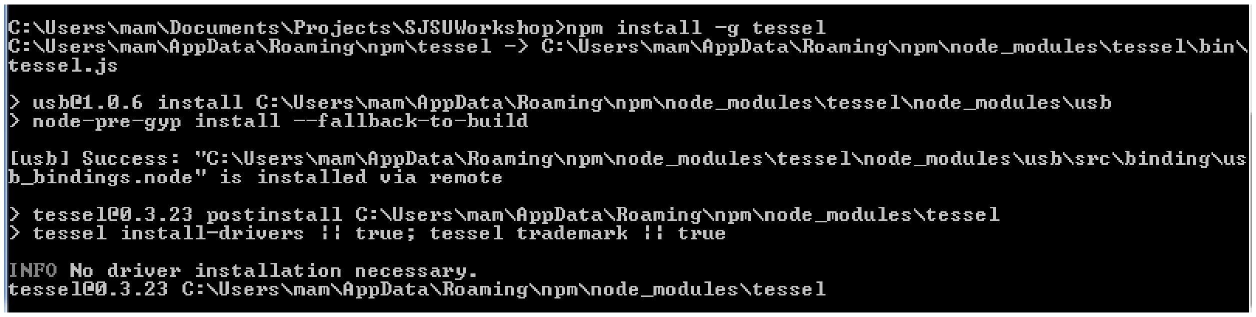
- Note: On Windows 7, you might encounter a "Driver not found" issue. In this case, you can go for the option to enable getting drivers from Windows Update that is under "Devices and Printers -> right-click on your computer name -> Device installation settings."
- If that doesn't work, the manual way to install the driver is to get Zadig208 and bind the Tessel to the WinUSB driver.

Step 2: Install Node JS from https://nodejs.org/en/download/. PLEASE NOTE: Tessel1 boards are compatible with Node version 0.12.7. Please make sure that the version downloaded matches 0.12.7.



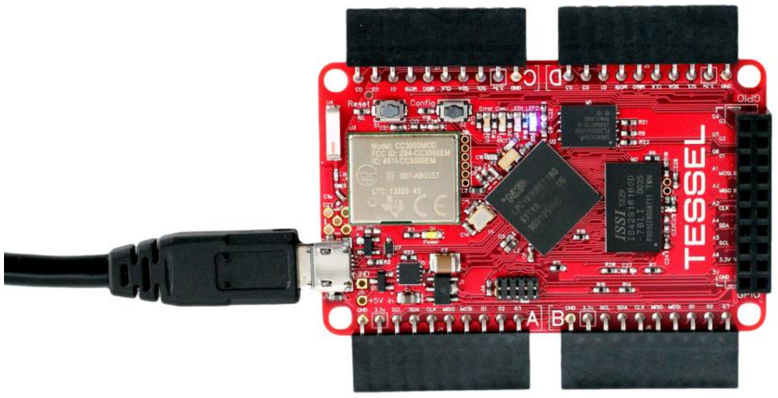
```
C:\Users\mam\Documents\Projects\SJSUWorkshop>node --version
v0.12.7
```

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:

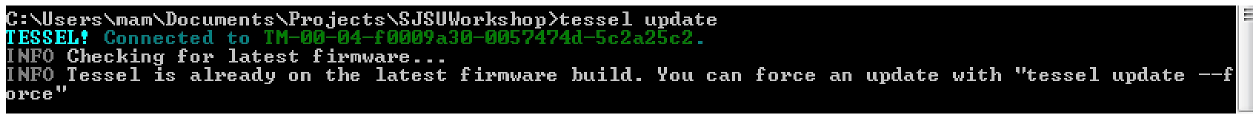


```
C:\Users\mam\Documents\Projects\SJSUWorkshop>npm install -g tessel
C:\Users\mam\AppData\Roaming\npm\tessel -> C:\Users\mam\AppData\Roaming\npm\node_modules\tessel\bin\
tessel.js

> 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"
}
```

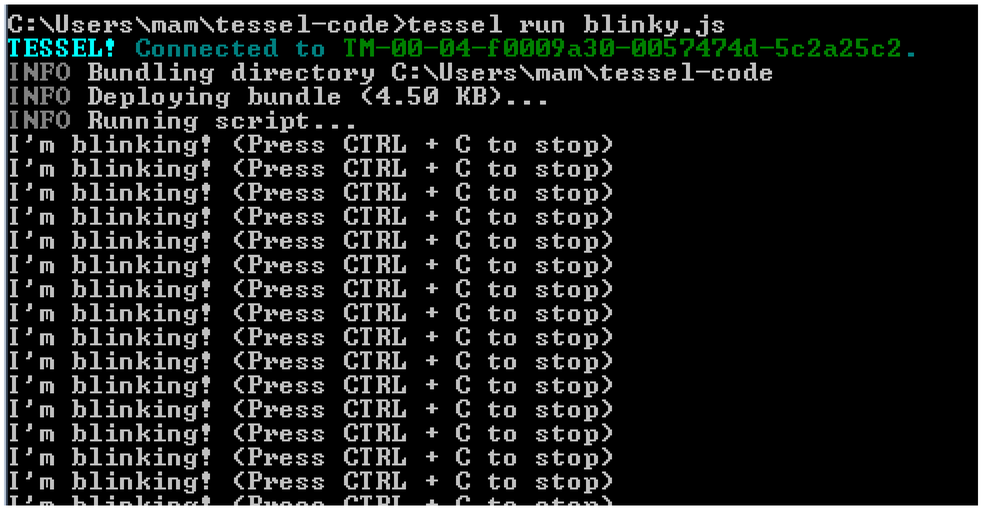
d. Open Notepad. Copy and paste the following, and save the file as – blinky.js in the Tessel-code folder.

```
// 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 the Tessel-code folder and type the commands:

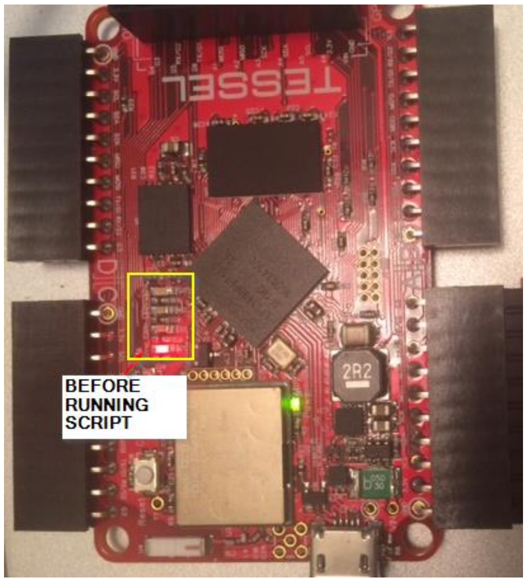
1. `npm init -y`
2. `tessel run blinky.js`

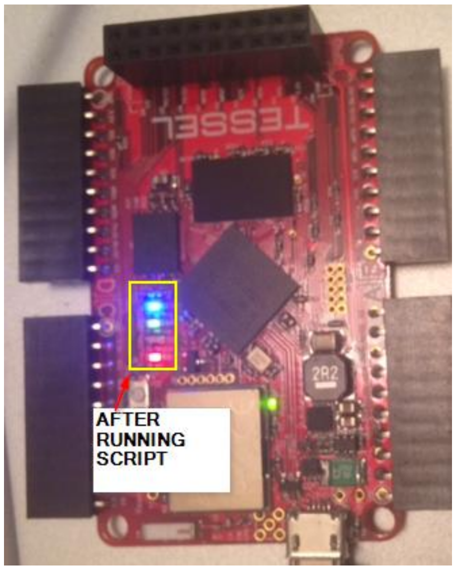
In the terminal, you will see the following output. Press Ctrl+C to stop the script.



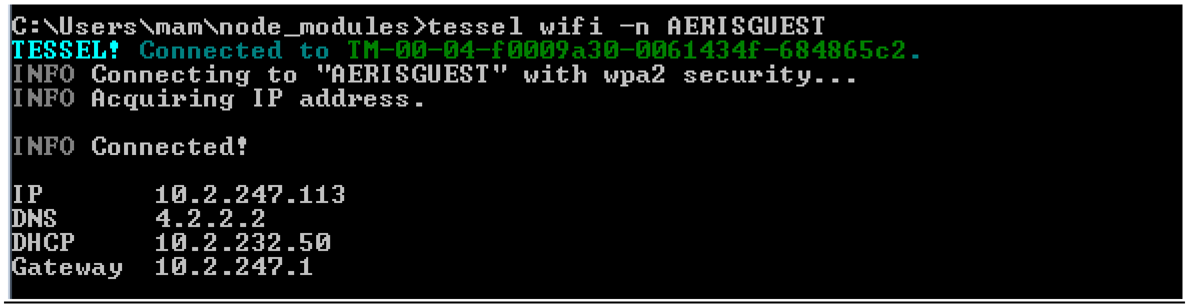
```
C:\Users\mam\tessel-code>tessel run blinky.js
TESSEL! Connected to TM-00-04-f0009a30-0057474d-5c2a25c2.
INFO Bundling directory C:\Users\mam\tessel-code
INFO Deploying bundle (4.50 KB)...
INFO Running script...
I'm blinking! (Press CTRL + C to stop)
I'm blinking! (Press CTRL + C to stop)
I'm blinking! (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: Set up and test the Wi-Fi connection.
Check your Wi-Fi connection: If the security type is = unsecured (like AERISGUEST)
In command prompt: Type the command: **tessel wifi –n AERISGUEST**



```
C:\Users\mam\node_modules>tessel wifi -n AERISGUEST
TESSEL! Connected to TM-00-04-f0009a30-0061434f-684865c2.
INFO Connecting to "AERISGUEST" with wpa2 security...
INFO Acquiring IP address.

INFO Connected!

IP        10.2.247.113
DNS       4.2.2.2
DHCP      10.2.232.50
Gateway   10.2.247.1
```

If you run into errors: First, check to make sure the yellow status light is on. If it is not, you are not connected to Wi-Fi. Try power-cycling your Tessel, and then run the **tessel wifi** command again.

Check your Wi-Fi connection: If the security type is = wpa2
In the terminal, type the command:

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

   In the terminal, you will see this:

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

IP 192.167.0.132
DNS 192.167.0.1
DHCP 192.167.0.1
Gateway 192.167.0.1
```

   This means you are successfully connected to Wi-Fi.

b) Test the Wi-Fi connection (optional – check your Wi-Fi connection: if security type is = wpa2).
   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 Tessel-code 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. Run the command: `tessel run wifi.js`. The following will be the output in the terminal:

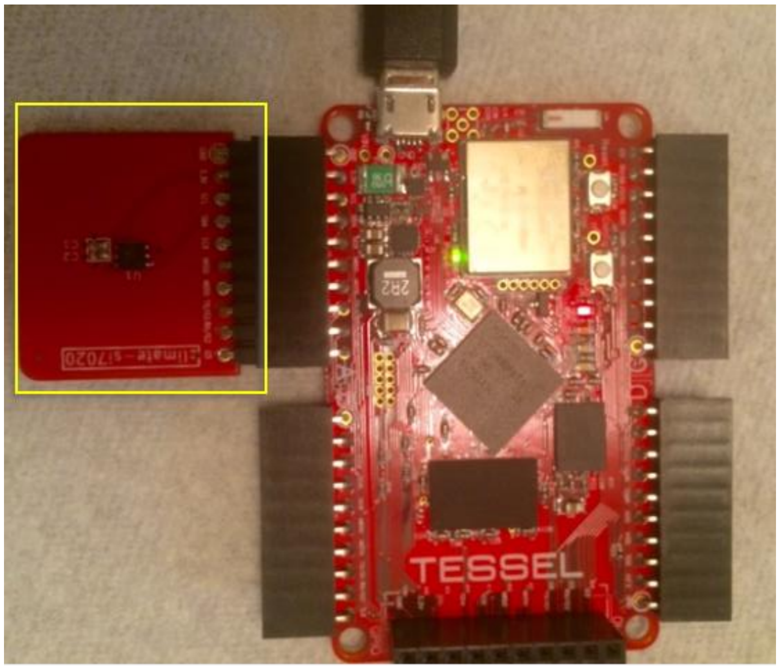
```
TESSEL! Connected to TM-00-04-f000da30-00624f54-126565c2.
INFO Bundling directory /Users/maanasamadiraju/tessel-code/wifi
INFO Deploying bundle (5.50 KB)...
INFO Running script...
http request #1
# statusCode 200
# received 200 OK
done.
```

END OF TASK 3

IV. READ TEMPERATURE FROM CLIMATE MODULE AND SEND TO AERCLOUD

INSTRUCTIONS:

Step 1: Connect Climate module to **PORT – A** of the Tessel board that is connected to your computer and has Wi-Fi 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. Then save the climate.js file.

Important: *Plug in your account number and API Key from Task 1 in place of <accountId> and <your-api-key> 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.
```

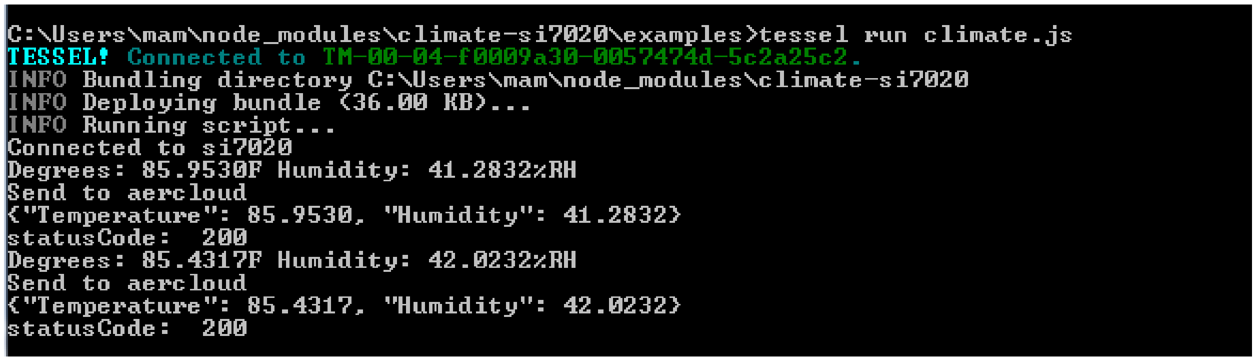
```
********************************************/

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/<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);
    });
}
```

Explanation: The above code basically reads the temperature and humidity values given by climate module and posts this data to the AerCloud container created in Task 2.

The flow is as follows:

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

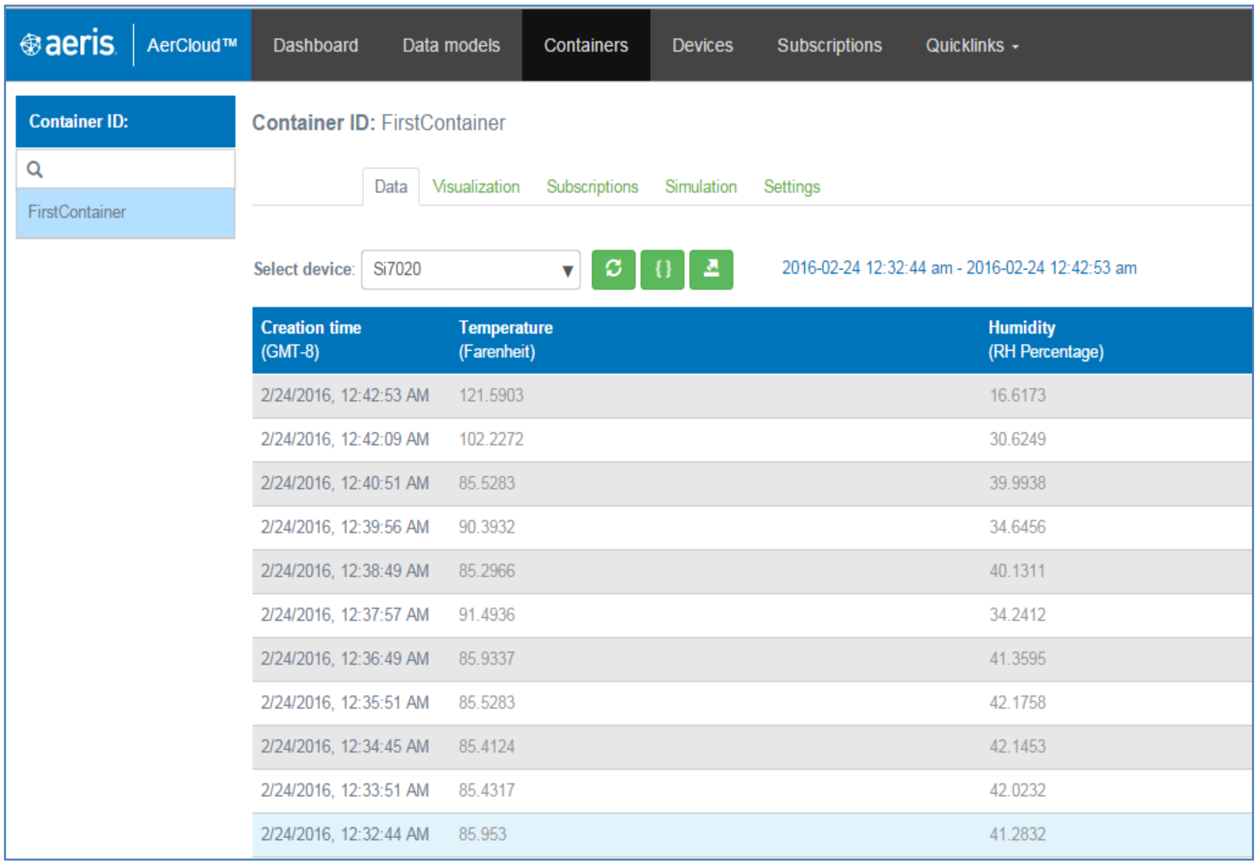
In the terminal, you will see the values being read and being sent to AerCloud at every 1-minute interval.



```
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. Log into AerCloud. Click on the “Container” tab from the top navigation 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

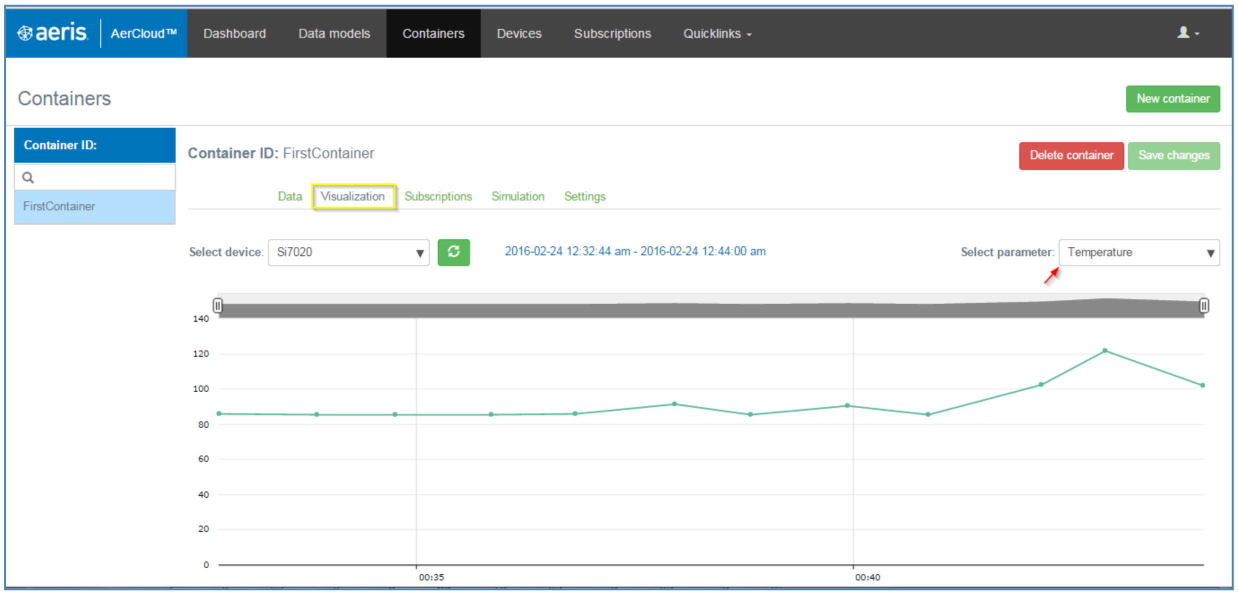
Select device: Si7020

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" as a graphical representation on how Temperature and Humidity are varying.

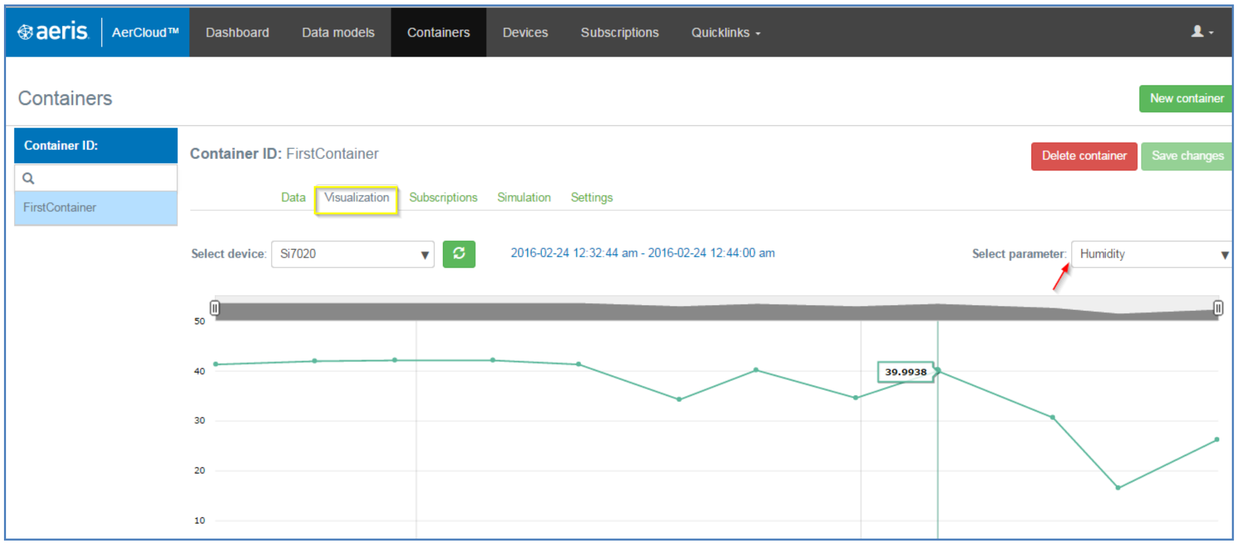
GRAPHICAL REPRESENTATION OF TEMPERATURE VARIATION:



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. WRITE DATA STREAM FROM AERCLOUD TO .CSV FILE

The objective of this task is to be able to access data from AerCloud and write to an external location. Specifically, we will 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.

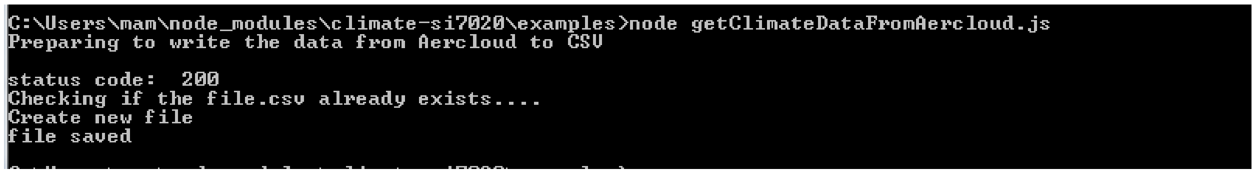
Step 1: Open Notepad and copy-paste the below code and save it as “getClimateDataFromAercloud.js” in /climate-Si7020/examples folder. Please note to enter your account ID and API Key.

```
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;}}
```

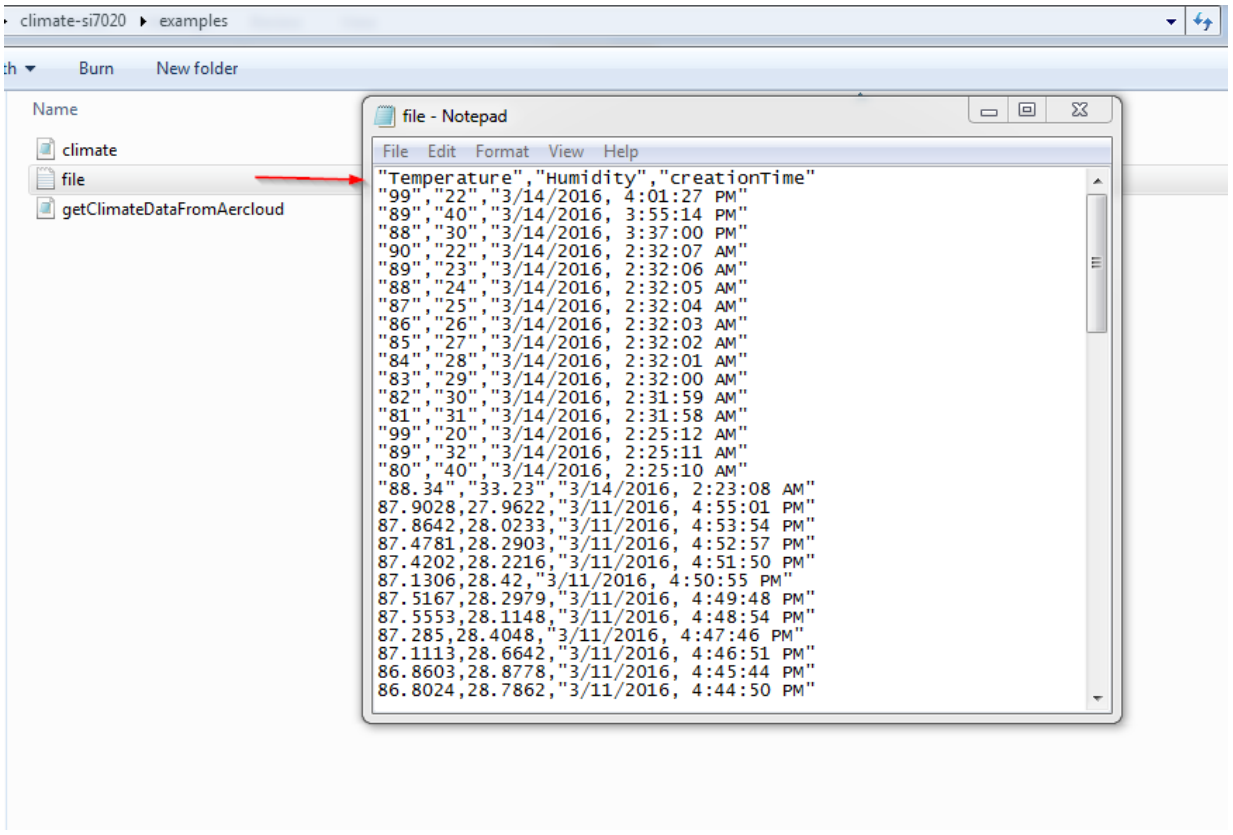
Step 2: Open Command prompt and navigate to the /climate-si7020/examples folder (the location where you saved the above file).

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 file called “file.csv” has been created.

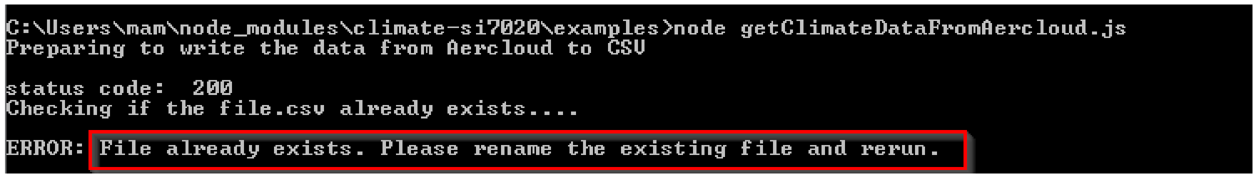


climate-si7020 ▸ examples
Burn
New folder
Name
climate
file
getClimateDataFromAercloud
file - Notepad
File Edit Format View Help

```
"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 set that has been sent to AerCloud.

NOTE: If there is a file.csv in the folder, you will get an error. The resolution is to rename the existing file.csv as file_1.csv and RERUN the code. You will then see an 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