Introduction to IoT

Lab 8(oneM2M and OM2M)

1. Software Tools Required

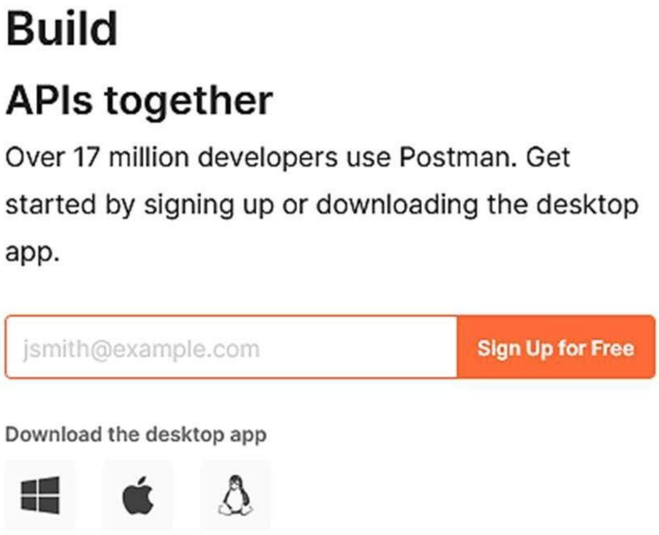
- Postman REST-API Client
- OM2M Server
- Java -JDK 1.8

2. Setup and Procedure

- For conducting this experiment, the below tools are necessary.
    1. Postman Rest client tool
    2. OM2M Server Setup.
- After installation, custom resources in om2m can be created through postman.

Installing Postman Rest Client:

- Go to the link: http://www.postman.com. On this page, you will see a section as given below.
- Click on the icon below Download the desktop app, you will be redirected to the download page, from here download the app according to your operating system.
- After downloading, install it. After installing, the postman homepage will appear.
  Note: Account Creation is not necessary can be skipped

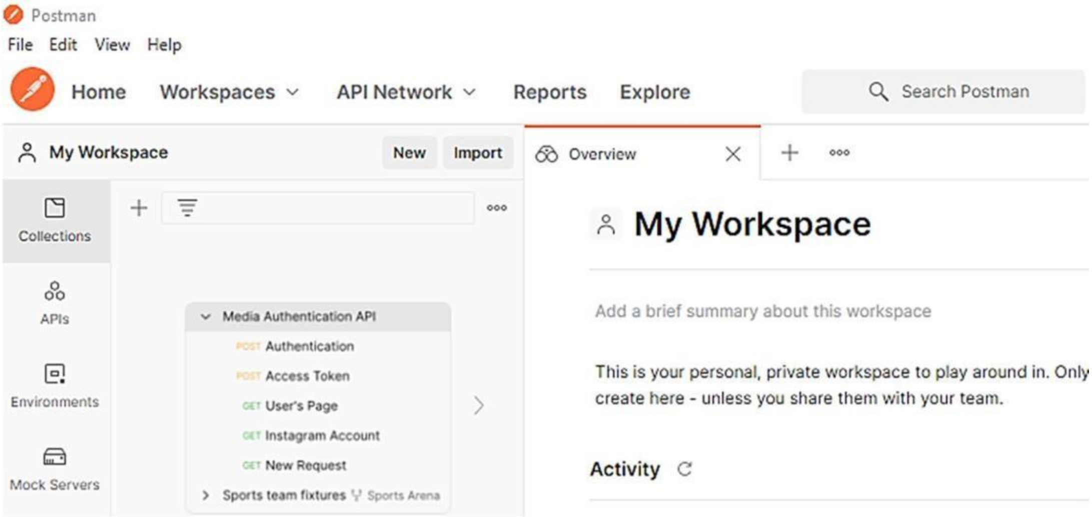


Build

APIs together

Over 17 million developers use Postman. Get started by signing up or downloading the desktop app.

Figure 1 Postman tool



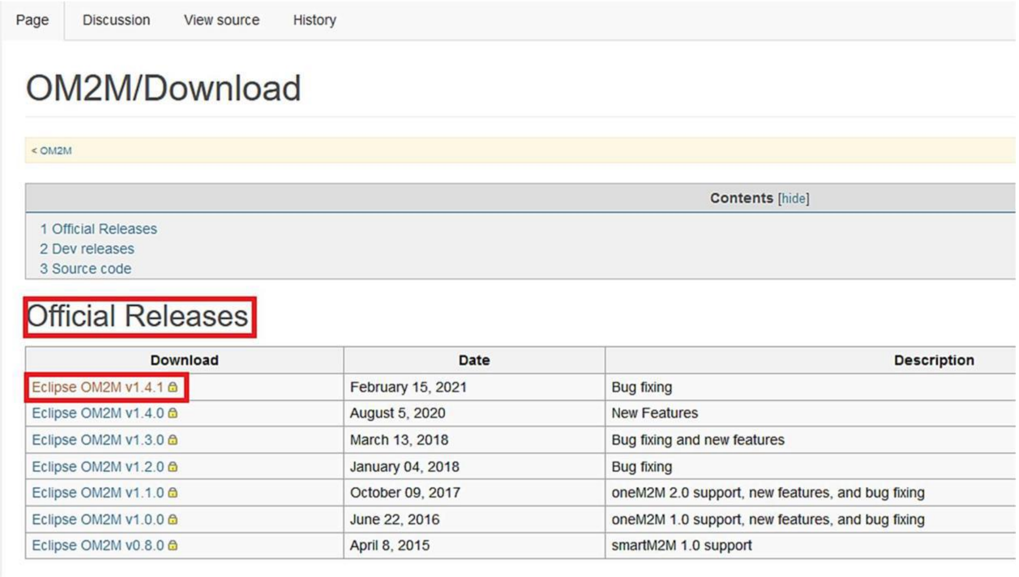
Add a brief summary about this workspace

This is your personal, private workspace to play around in. Only create here - unless you share them with your team.

Figure 2 Postman Main Screen

OM2M Server Setup

- Go to the link: http://wiki.eclipse.org/OM2M/Download.
- Click on the OM2M 1.4.1 Package to download the latest release as shown in fig. below.
- After Successful download, extract the zip file "eclipse-om2m-v1-4-1.zip".



OM2M/Download

< OM2M

Contents [hide]

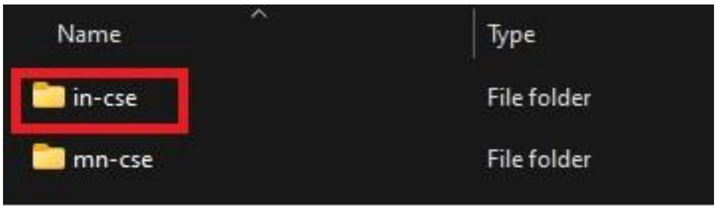
1 Official Releases
2 Dev releases
3 Source code

Official Releases

Download	Date	Description
Eclipse OM2M v1.4.1	February 15, 2021	Bug fixing
Eclipse OM2M v1.4.0	August 5, 2020	New Features
Eclipse OM2M v1.3.0	March 13, 2018	Bug fixing and new features
Eclipse OM2M v1.2.0	January 04, 2018	Bug fixing
Eclipse OM2M v1.1.0	October 09, 2017	oneM2M 2.0 support, new features, and bug fixing
Eclipse OM2M v1.0.0	June 22, 2016	oneM2M 1.0 support, new features, and bug fixing
Eclipse OM2M v0.8.0	April 8, 2015	smartM2M 1.0 support

Figure 3 OM2M Download Page

- There are two folders named "in-cse" and "mn-cse".
- Go to the "in-cse" folder and inside the "configuration" folder open "config.ini" file with notepad and in line 5 change the port to 5089.



Name	Type
in-cse	File folder
mn-cse	File folder

Figure 4 in-cse folder

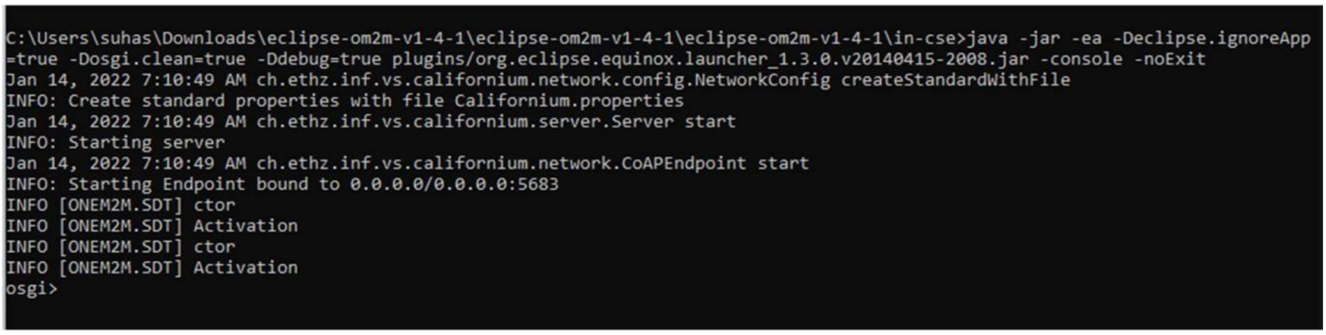
- Now go to Command prompt (in Windows search for "cmd" in start, and open) or Terminal (Open a Linux Terminal Using Ctrl + Alt + T or search Terminal in Spotlight search in Mac OS).
- Once cmd/terminal opens enter the command "**java–version**".
- Make sure you are in the right java–version since OM2M will only be compatible with jdk 1.8.0_X versions.
- In case if you need to install jdk, please refer to the below URLs based on the OS:

Windows: https://codenotfound.com/java-download-install-jdk-8-windows.html

Linux: https://docs.datastax.com/en/jdk-install/doc/jdk-install/installOpenJdkDeb.html

Mac: https://stackoverflow.com/questions/24342886/how-to-install-java-8-on-mac

- Now in cmd/terminal enter start.bat/start.sh to launch the OM2M in-cse interface. (Double click generally works as well inside the contents of in-cse folder). Wait for the java command to get executed.
- Once the OM2M launches successfully, go to the OM2M login page using the link below, a page as given will appear. Link: http://127.0.0.1:5089/webpage



```
C:\Users\suhas\Downloads\eclipse-om2m-v1-4-1\eclipse-om2m-v1-4-1\eclipse-om2m-v1-4-1\in-cse>java -jar -ea -Declipse.ignoreApp
=true -Dosgi.clean=true -Ddebug=true plugins/org.eclipse.equinox.launcher_1.3.0.v20140415-2008.jar -console -noExit
Jan 14, 2022 7:10:49 AM ch.ethz.inf.vs.californium.network.config.NetworkConfig createStandardWithFile
INFO: Create standard properties with file Californium.properties
Jan 14, 2022 7:10:49 AM ch.ethz.inf.vs.californium.server.Server start
INFO: Starting server
Jan 14, 2022 7:10:49 AM ch.ethz.inf.vs.californium.network.CoAPEndpoint start
INFO: Starting Endpoint bound to 0.0.0.0/0.0.0.0:5683
INFO [ONEM2M.SDT] ctor
INFO [ONEM2M.SDT] Activation
INFO [ONEM2M.SDT] ctor
INFO [ONEM2M.SDT] Activation
osgi>
```

Figure 5 OM2M Successfully launched

- Enter the default username and password as admin, admin, and press login. We can see a default resource tree of OM2M as given below.

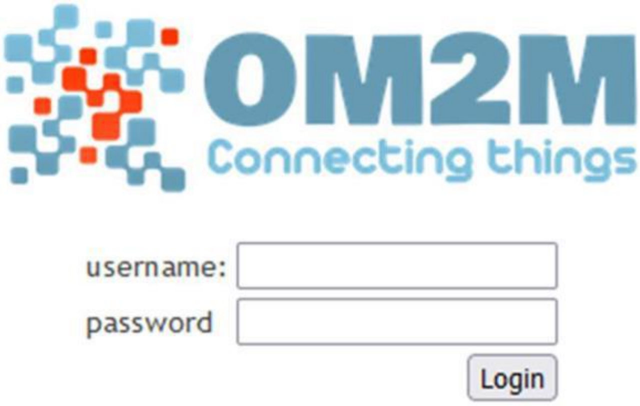
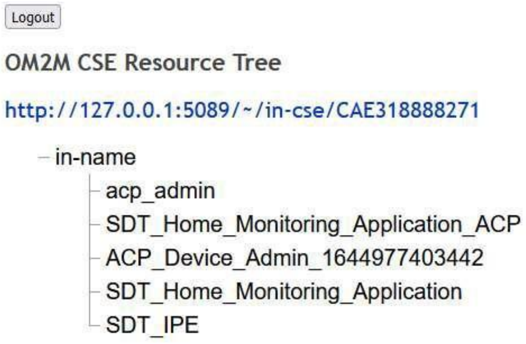


Figure 6 OM2M Login page

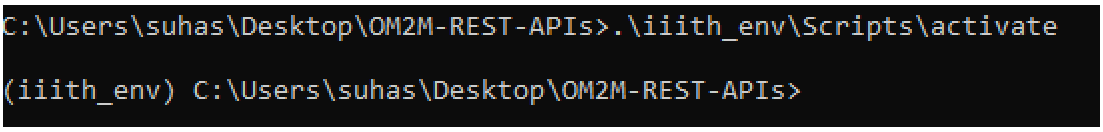


```
Logout

OM2M CSE Resource Tree

http://127.0.0.1:5089/~/in-cse/CAE318888271

 - in-name
     acp_admin
     SDT_Home_Monitoring_Application_ACP
     ACP_Device_Admin_1644977403442
     SDT_Home_Monitoring_Application
     SDT_IPE
```

Figure 7 Default OM2M Resource Tree

Python and Jupyter notebook Setup

- The instructions for installing Python can be found at https://realpython.com/installing-python/
- Once the python has been setup install the virtual environment and activate the virtual environment https://packaging.python.org/en/latest/guides/installing-using-pip-and-virtual-environments/ (for windows you can use "**py -m venv iiith_env**" and "**.\iiith_env\Scripts\activate**" to activate it)



```
C:\Users\suhas\Desktop\OM2M-REST-APIs>.\iiith_env\Scripts\activate

(iiith_env) C:\Users\suhas\Desktop\OM2M-REST-APIs>
```

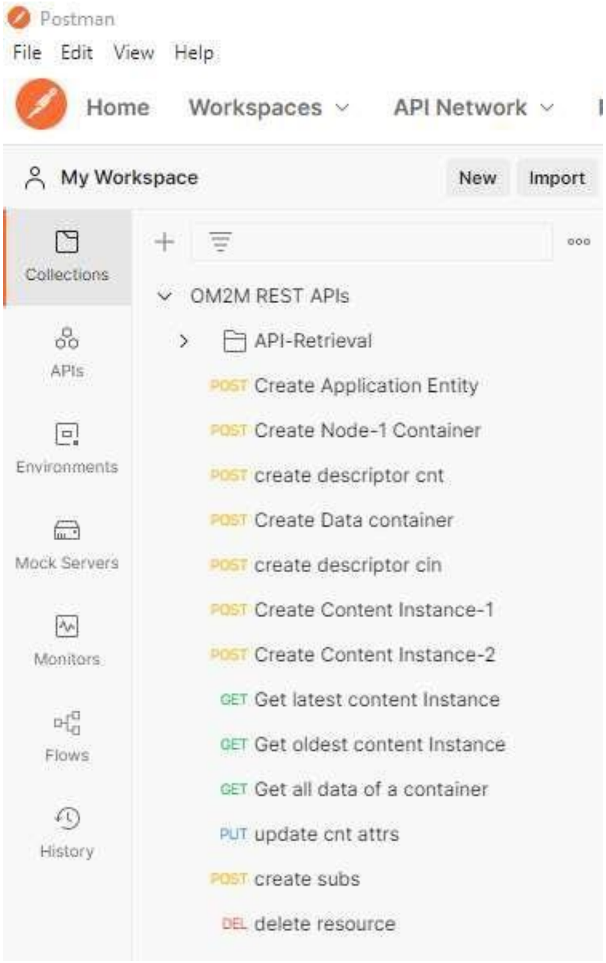
Figure 8 Activated Virtual Environment

- Now install the requirements file by using the command "**pip install -r requirements.txt**" for windows users and "**pip3 install -r requirements.txt**" for

3. Part 1: Creating the OM2M Resource Tree using Postman.

- Go to the Postman then click Workspaces then press Import then Upload Files and select the JSON files named as ‘OM2M REST APIs.postman_collection.json’
- Click on Import. We can see the request collections below the OM2M Rest APIs
- Before using the collection make sure the OM2M server is running and available at http://127.0.0.1:5089/webpage. The request can be sent by pressing the “Send” button in postman, after selecting any request.
- Now send each request one by one to construct the resource tree.

- We can notice the changes in the OM2M server after each request by clicking on Tree node, and notice the headers, body, and responses after each request in Postman. Following are the requests to be sent in steps,

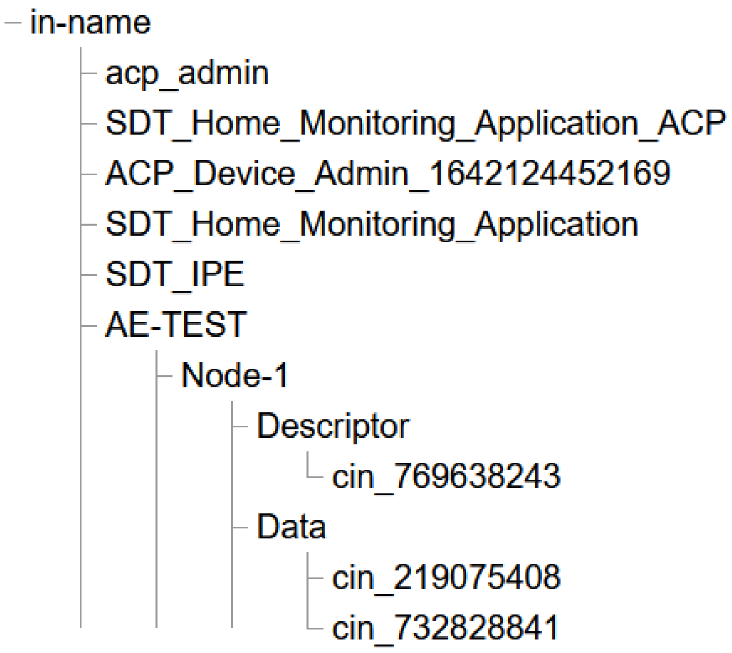


OM2M REST APIs
- API-Retrieval
- POST Create Application Entity
- POST Create Node-1 Container
- POST create descriptor cnt
- POST Create Data container
- POST create descriptor cin
- POST Create Content Instance-1
- POST Create Content Instance-2
- GET Get latest content Instance
- GET Get oldest content Instance
- GET Get all data of a container
- PUT update cnt attrs
- POST create subs
- DEL delete resource

Figure 9 OM2M Postman Collection

1. Creating an Application Entity: Select Post Create Application Entity Press Send
2. Creating a container for device called Node-1: Select Post Create Node-1 container Press Send
3. Creating a Descriptor container: Select Post create descriptor cnt Press Send
4. Creating a Data container: Select Post Create Data Container Press Send

5. Creating descriptor content instance: Select Post create descriptor cin Press Send
6. Creating data content instance 1 & 2 : Select Post Create Content instance 1/2 Press Send
7. Retrieving the latest content instance: Select GET latest Content instance Press Send
8. Retrieving the oldest content instance: Select GET oldest Content instance Press Send
9. Retrieving all content instances: Select GET all data of a container Press Send
10. Updating number of attributes inside data container: Select PUT update cnt attrs □ Send



```
– in-name
   ├ acp_admin
   ├ SDT_Home_Monitoring_Application_ACP
   ├ ACP_Device_Admin_1642124452169
   ├ SDT_Home_Monitoring_Application
   ├ SDT_IPE
   ├ AE-TEST
   │  ├ Node-1
   │  │  ├ Descriptor
   │  │  │  └ cin_769638243
   │  │  ├ Data
   │  │  │  ├ cin_219075408
   │  │  │  └ cin_732828841
```

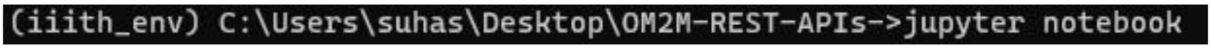
Figure 10 OM2M Resource Tree after Sending the requests

Optional (for deleting the resource):

11. Deleting the resource: Select DEL delete resource Press Send

4. Part-2: Publishing data to OM2M

- There are two approaches that can be followed for publishing data to OM2M.
   1. Using a Mock Device through Jupyter Notebook
   2. Using an ESP 32 Micro-Controller
- Now enter "**jupyter notebook**" in the cmd as shown below



```
(iiith_env) C:\Users\suhas\Desktop\OM2M-REST-APIs->jupyter notebook
```

Figure 11 Launching Jupyter Server Home page

Part 2 A: Mock Device through Jupyter Notebooks

- Make sure the OM2M is up and running for this part of the experiment.

- After launching the Jupyter notebook you can see the home page of Jupyter server as below.

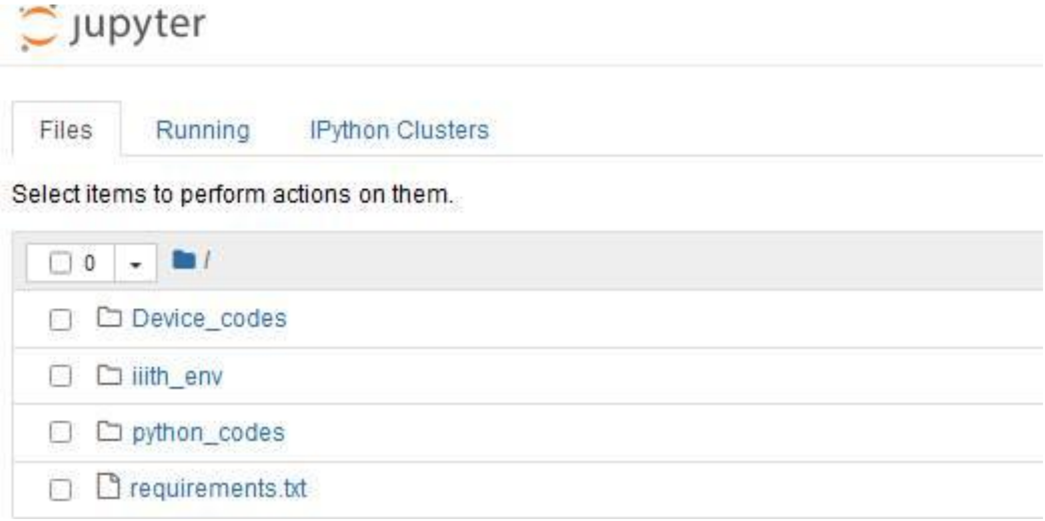


Figure 12 Jupyter Server Home page

- Now go to the folder “**Device_codes**” folder and open the notebook file “**Mock_Device.ipynb**”
- In the file click on “**Cell**” option and then press “**Run All**” to run all the cells in the notebook.

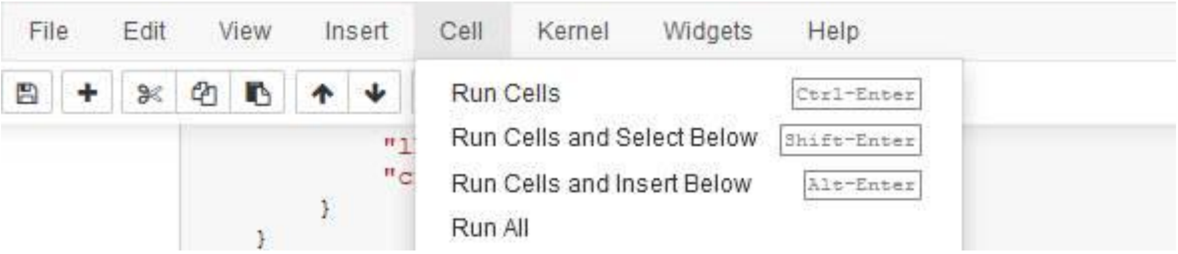
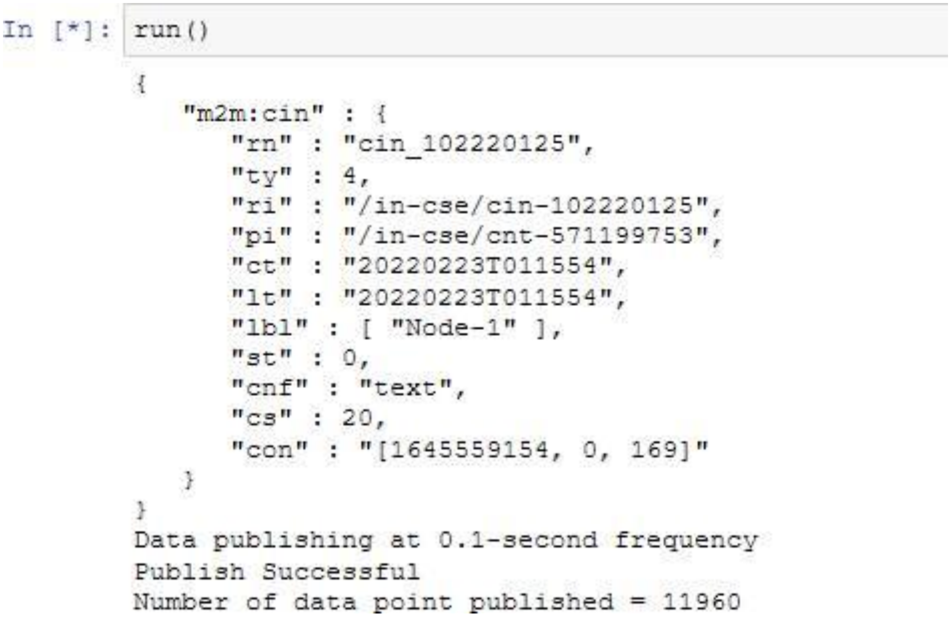


Figure 13 Run all Cells option

- The broad steps in this Notebook are as below
  1. Import required Python modules
  2. Define the default execution constants
  3. Create a OM2M Resource tree(if not exists)
  4. Define a random data publishing function
  5. Run the function in an infinite loop
- Now if everything goes well you see the last cell’s output as below.



```
In [*]: run()

{
    "m2m:cin" : {
        "rn" : "cin_102220125",
        "ty" : 4,
        "ri" : "/in-cse/cin-102220125",
        "pi" : "/in-cse/cnt-571199753",
        "ct" : "20220223T011554",
        "lt" : "20220223T011554",
        "lbl" : [ "Node-1" ],
        "st" : 0,
        "cnf" : "text",
        "cs" : 20,
        "con" : "[1645559154, 0, 169]"
    }
}
Data publishing at 0.1-second frequency
Publish Successful
Number of data point published = 11960
```

Figure 14 Mock Jupyter Device publishing data to OM2M

- The same data can be seen in the OM2M page under AE-TEST application entity, in the Data container of Node-1.

Part 2 B: Publishing Data from ESP-Micro-Controller

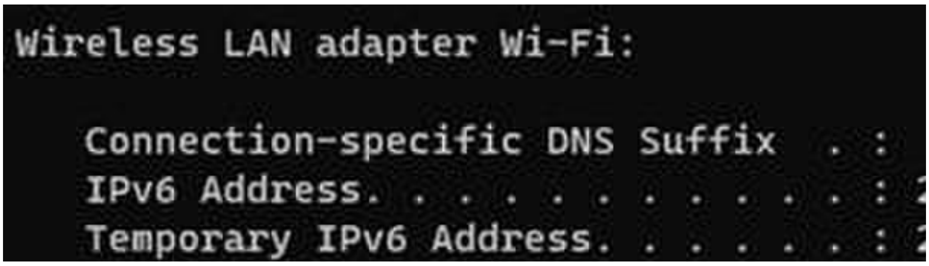
- There are two main pre-conditions that needs to be checked before using the Arduino sketch
  1. System IP (To access OM2M from other devices)
  2. Application entity with account name

System IP:

- To check the System IP (choose the external Ip instead of 127.0.0.1), choose the ipv4

  a. open "command prompt - cmd" in windows and enter "**ipconfig**"

  b. open terminal and enter "**ifconfig**" in the case of Linux/Mac



```
Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   IPv6 Address. . . . . . . . . . . :
   Temporary IPv6 Address. . . . . . :
```

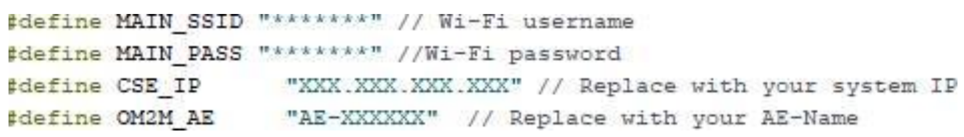
Figure 15 Example IP address-192.168.210.225

Appropriate Application Entity

- Using Postman create an application entity, node-1 container, descriptor container, data container for this experiment.

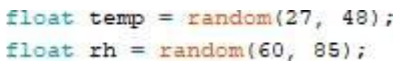
- The code can be broadly divided into the components below.
    1. Including the required header-file and create the object of the header file
    2. Defining the om2m parameters like host Ip, port number, mn, ae, data container.
    3. Connect to Wi-Fi
    4. Constructing the URL based on the om2m parameters
    5. Attaching the headers before sending the request
    6. Constructing the request with the data values of all the two sensor values
    7. Publishing the Data

- Now the following changes need to be made in the Publishing code to function
    1. SSID – Wi-Fi username
    2. Password- Wi-Fi password
    3. System IP- The system Ip needs to be used in the code
    4. AE-Name for publishing to correct AE
    5. Data content instance value for publishing appropriate values



```
#define MAIN_SSID "*******" // Wi-Fi username
#define MAIN_PASS "*******" //Wi-Fi password
#define CSE_IP      "XXX.XXX.XXX.XXX" // Replace with your system IP
#define OM2M_AE     "AE-XXXXXX"  // Replace with your AE-Name
```

Figure 16 Parameter Changes in the Code



```
float temp = random(27, 48);
float rh = random(60, 85);
```

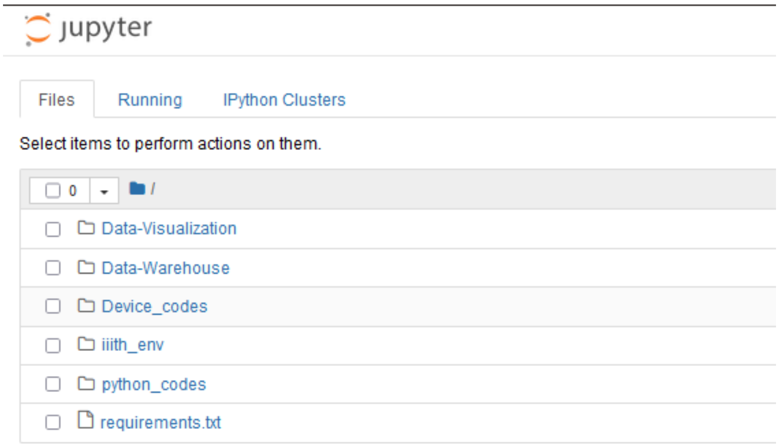
Figure 17 Replace with DHT Sensor functions

- Once the Serial monitor prints a 201, the data gets published, and the same data can be seen on OM2M.

5. Part-3: Visualization of data on OM2M

- The below are the below steps for implanting a Simple Visualizing Script
    1. Get all data from OM2M.
    2. Clean all data for retrieving the actual content.
    3. Plot the data.

- Make sure the OM2M is up and running for this part of the experiment.
- Follow the instructions of part 2 for publishing data
- Make sure the data is continuously posting to OM2M Server
- After launching the Jupyter notebook you can see the home page of Jupyter server as below



Select items to perform actions on them.

□ 0 ▾ /

□ Data-Visualization
□ Data-Warehouse
□ Device_codes
□ iiith_env
□ python_codes
□ requirements.txt

Figure 18 Code files

- Now go to "**Data-Visualization**" folder and open the notebook file "**Simple Dynamic Plots.ipynb**"
- In the file click on "**Cell**" option and then press "**Run All**" to run all the cells in the notebook.

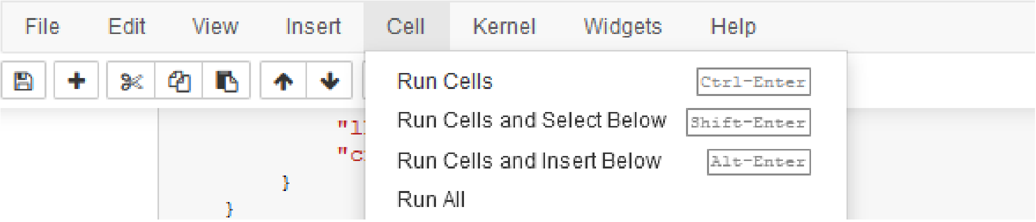

File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Run Cells Ctrl-Enter
Run Cells and Select Below Shift-Enter
Run Cells and Insert Below Alt-Enter
Run All

Figure 19 Run all cells option.

- The broad Steps involved in this script are as below.
  1. Import required Modules.
  2. Define a function for
     a. Retrieving all data from OM2M
     b. Cleaning the data
     c. Generating plots for data
  3. Running this function at 1 second interval
- Now if everything went well you should be able to see a live graph.