

## CPE 322 – Lab 7

1. First, I made an account on ThingSpeak.
2. Then I made a new channel:

### New Channel

Name	<input type="text" value="cpu_loop"/>
Description	<input type="text" value="CPE 322 - Lab 7"/>
Field 1	<input type="text" value="cpu_pc"/> <input checked="" type="checkbox"/>
Field 2	<input type="text" value="mem_avail_mb"/> <input checked="" type="checkbox"/>

a.

3. I then ran the following code in Ubuntu:

```
jstefan1@DESKTOP-6SV6DJ0:~/iot/lesson6$ cd ~/demo
jstefan1@DESKTOP-6SV6DJ0:~/demo$ cp ~/iot/lesson7/thingspeak_cpu_loop.py .
jstefan1@DESKTOP-6SV6DJ0:~/demo$ cp ~/iot/lesson7/thingspeak_feed.py .
jstefan1@DESKTOP-6SV6DJ0:~/demo$ cat thingspeak_cpu_loop.py
import http.client, urllib.request, urllib.parse, urllib.error
from time import localtime, strftime
import psutil
import time
def doit():
    cpu_pc = psutil.cpu_percent()
    mem = psutil.virtual_memory()
    mem_avail_mb = mem.available/(1024*1024)
    params = urllib.parse.urlencode({'field1': cpu_pc, 'field2':
        mem_avail_mb, 'key': 'YOURKEYHERE'})
    headers = {"Content-type":
        "application/x-www-form-urlencoded", "Accept": "text/plain"}
    conn = http.client.HTTPConnection("api.thingspeak.com:80")
    try:
        conn.request("POST", "/update", params, headers)
        response = conn.getresponse()
        print(cpu_pc)
        print(mem_avail_mb)
        print(strftime("%a, %d %b %Y %H:%M:%S", localtime()))
        print(response.status, response.reason)
        data = response.read()
        conn.close()
    except:
        print("connection failed")
    #sleep for 60 seconds (message update interval limit of 15 seconds)
if __name__ == "__main__":
```

a.

```
jstefan1@DESKTOP-6SV6DJ0:~/demo$ cat thingspeak_feed.py
import http.client, urllib.request, urllib.parse, urllib.error
from time import localtime, strftime
import psutil
import time
"""
2020-11-26: Pridhvi Myneni added lines 13-18, 25, and 43-54
1. gitignore (https://git-scm.com/docs/gitignore) now prevents students from pushing up credentials.
2. This program now attempts to cache the API key, but always prompts the user for their preference prior to saving credentials locally.
3. Pickle data type used to store credentials. Pickle is a binary storage version of JSON. Please note not-human-readable does not mean secure.
4. Removed key storage in source, which is generally a bad practice, as it leads to version control systems having your key.
5. Key for storage in a dictionary. This could be expanded in the future if other information needs to be similarly stored.
"""
from pickle import dump as pickle_dump
from pickle import load as pickle_load
from os.path import exists as file_exists
from os.path import isfile as is_file
PICKLE_FILE_NAME = "API_KEY.pickle"
API_KEY_INDEX = "API_KEY"
def doit():
    cpu_pc = psutil.cpu_percent()
    mem = psutil.virtual_memory()
    mem_avail_mb = mem.available/(1024*1024)
    params = urllib.parse.urlencode({'field1': cpu_pc, 'field2':
#         mem_avail_mb, 'key': 'YOURKEYHERE'})
    mem_avail_mb, 'key': KEY})
```

b.

```
jstefan1@DESKTOP-6SV6DJ0:~/demo$ python3 thingspeak_feed.py
An API key savefile was not found. Enter Write API Key >>>
Should we save this key for future use? [y/N] >>> y
```

c.

```

jstefan1@DESKTOP-6SV6DJ0:~/demo$ python3 thingspeak_feed.py
An API key savefile was not found. Enter Write API Key >>> W
Should we save this key for future use? [y/N] >>> y
0.0
2992.0546875
Tue, 07 Nov 2023 18:59:32
200 OK
0.1
2990.20703125
Tue, 07 Nov 2023 19:00:33
200 OK
0.1
2989.94921875
Tue, 07 Nov 2023 19:01:33
200 OK
0.0
2989.97265625
Tue, 07 Nov 2023 19:02:33
200 OK
0.1
2989.58203125
Tue, 07 Nov 2023 19:03:33
200 OK
0.1
2989.48828125
Tue, 07 Nov 2023 19:04:33
200 OK
0.1
2989.6328125

```

- d.
4. I then signed into Google Cloud Platform Identity and Access Management and made a new project called cpudata. I went into my APIs and Service and enabled Google Drive API and Google Sheets API. I then created a service account under credentials called cpudata. I then created a new JSON key and it downloaded the JSON file to my computer.
5. I then installed gspread and oauth2client:

```
jstefan1@DESKTOP-6SV6DJ0:~$ sudo pip3 install -U gspread oauth2client
Collecting gspread
  Downloading gspread-5.12.0-py3-none-any.whl (48 kB)
    48.5/48.5 KB 440.0 kB/s eta 0:00:00
Collecting oauth2client
  Downloading oauth2client-4.1.3-py2.py3-none-any.whl (98 kB)
    98.2/98.2 KB 2.4 MB/s eta 0:00:00
Collecting google-auth-oauthlib<=0.4.1
  Downloading google_auth_oauthlib-1.1.0-py2.py3-none-any.whl (19 kB)
Collecting google-auth>=1.12.0
  Downloading google_auth-2.23.4-py2.py3-none-any.whl (183 kB)
    183.3/183.3 KB 1.6 MB/s eta 0:00:00
Collecting pyasn1-modules>=0.0.5
  Downloading pyasn1_modules-0.3.0-py2.py3-none-any.whl (181 kB)
    181.3/181.3 KB 3.9 MB/s eta 0:00:00
Collecting pyasn1>=0.1.7
  Downloading pyasn1-0.5.0-py2.py3-none-any.whl (83 kB)
    83.9/83.9 KB 5.7 MB/s eta 0:00:00
Requirement already satisfied: httplib2>=0.9.1 in /usr/lib/python3/dist-packages (from oauth2client) (0.20.2)
Requirement already satisfied: six>=1.6.1 in /usr/lib/python3/dist-packages (from oauth2client) (1.16.0)
Collecting rsa>=3.1.4
  Downloading rsa-4.9-py3-none-any.whl (34 kB)
Collecting cachetools<6.0,>=2.0.0
  Downloading cachetools-5.3.2-py3-none-any.whl (9.3 kB)
Collecting requests-oauthlib>=0.7.0
  Downloading requests_oauthlib-1.3.1-py2.py3-none-any.whl (23 kB)
Requirement already satisfied: pyparsing!=3.0.0,!3.0.1,!3.0.2,!3.0.3,<4,>=2.4.2 in /usr/lib/python3/dist-packages (from httplib2>=0.9.1->oauth2client) (2.4.7)
Requirement already satisfied: oauthlib>=3.0.0 in /usr/lib/python3/dist-packages (from requests-oauthlib>=0.7.0->google-auth-oauthlib>=0.4.1->gspread) (3.2.0)
```

a.

6. Then I ran the following commands:

```
jstefan1@DESKTOP-6SV6DJ0:~$ cd demo
jstefan1@DESKTOP-6SV6DJ0:~/demo$ cp ~/iot/lesson3/system_info.py .
jstefan1@DESKTOP-6SV6DJ0:~/demo$ cp ~/iot/lesson7/cpu_spreadsheet.py .
jstefan1@DESKTOP-6SV6DJ0:~/demo$ sudo cp /mnt/c/Users/joeys/Downloads/cpdata-404500-82f06f46c598.json .
```

a.

b.

7. I then made a new Google Sheets called cpudata and shared it with the client email that was in the JSON file. I then deleted rows 2-1000 since they are technically blank rows that will not be overwritten.

8. I then ran these commands. For the nano command, I changed the JSON file to match the one I had.

```
jstefan1@DESKTOP-6SV6DJ0:~/demo$ nano cpu_spreadsheet.py
jstefan1@DESKTOP-6SV6DJ0:~/demo$ python3 cpu_spreadsheet.py
Logging sensor measurements to cpudata every 10 seconds.
Press Ctrl-C to quit.
/home/jstefan1/demo/cpu_spreadsheet.py:31: DeprecationWarning: [Deprecated][in version 6.0.0]: client_factory will be replaced by gspread.http_client types
  worksheet = login_open_sheet(GDOCS_OAUTH_JSON, GDOCS_SPREADSHEET_NAME)
```

a.

b.

```
jstefan1@DESKTOP-6SV6DJ0:~/demo$ python3 cpu_spreadsheet.py
Logging sensor measurements to cpudata3 every 10 seconds.
Press Ctrl-C to quit.
/home/jstefan1/demo/cpu_spreadsheet.py:31: DeprecationWarning: [Deprecated][in version 6.0.0]: client_factory will be replaced by gspread.http_client types
  worksheet = login_open_sheet(GDOCS_OAUTH_JSON, GDOCS_SPREADSHEET_NAME)
2023-11-08 00:35:40.675101
CPU Usage: 2.5%
Memory Available: 3.0 GB
Wrote a row to cpudata3
2023-11-08 00:35:50.944414
CPU Usage: 0.0%
Memory Available: 3.0 GB
Wrote a row to cpudata3
2023-11-08 00:36:01.176310
CPU Usage: 0.1%
Memory Available: 3.0 GB
Wrote a row to cpudata3
2023-11-08 00:36:11.342127
CPU Usage: 0.0%
Memory Available: 3.0 GB
Wrote a row to cpudata3
2023-11-08 00:36:21.499777
CPU Usage: 0.0%
Memory Available: 3.0 GB
Wrote a row to cpudata3
2023-11-08 00:36:31.683434
CPU Usage: 0.1%
Memory Available: 3.0 GB
```

C.

cpudata3				
File Edit View Insert Format Data Tools Extensions Help				
<input type="text"/> <input type="button" value="↶"/> <input type="button" value="↷"/> <input type="button" value="🖨"/> <input type="button" value="📋"/> 100%           \$ % .0 .00 123   Default...				
D3	fx			
	A	B	C	D
1	Date / Time	CPU Usage %	Memory Available GB	
2	2023-11-08 00:3	2.5	2.957122803	
3	2023-11-08 00:3	0	2.955291748	
4	2023-11-08 00:3	0.1	2.954559326	
5	2023-11-08 00:3	0	2.954559326	
6	2023-11-08 00:3	0	2.954559326	
7	2023-11-08 00:3	0.1	2.954559326	
8	2023-11-08 00:3	0	2.954559326	
9	2023-11-08 00:3	0.1	2.954559326	
10	2023-11-08 00:3	0	2.954559326	

d.