

```
Roshans-MBP:lesson6 roshansoni$ sudo pip3 install -U psutil
WARNING: The directory '/Users/roshansoni/Library/Caches/pip' or its parent directory is not owned or is not writable by the current user. The cache has been disabled. Check the permissions and owner of that directory. If executing pip with sudo, you should use sudo's -H flag.
Requirement already satisfied: psutil in /Library/Frameworks/Python.framework/Versions/3.9/lib/python3.9/site-packages (5.9.0)
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
```

```
Roshans-MBP:~ roshansoni$ cd ~/roshangit/demo/
Roshans-MBP:demo roshansoni$ cp ~/roshangit/iot/lesson7/thingspeak_cpu_loop.py .
Roshans-MBP:demo roshansoni$ cp ~/roshangit/iot/lesson7/thingspeak_feed.py .
Roshans-MBP:demo roshansoni$ 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__":
    while True:
        doit()
        time.sleep(60)
```

```

Roshans-MBP:demo roshansoni$ 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})
    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__":
#caching the Write Key for the ThingSpeak API
    if file_exists(PICKLE_FILE_NAME) and is_file(PICKLE_FILE_NAME):
        save_file = None
        with open(PICKLE_FILE_NAME, 'rb') as f:
            save_file = pickle_load(f)
        KEY = save_file[API_KEY_INDEX]
    else:
        key = input("An API key savefile was not found. Enter Write API Key >>> ")
        should_save = input("Should we save this key for future use? [y/N] >>> ")
        KEY = key.strip()
        if len(should_save)>0 and should_save.lower().startswith("y"):
            with open(PICKLE_FILE_NAME, 'wb') as f:
                pickle_dump({API_KEY_INDEX: KEY}, f)
    while True:
        doit()
        time.sleep(60)

```

```

[Roshans-MBP:demo roshansoni$ python3 thingspeak_feed.py
An API key savefile was not found. Enter Write API Key >>> R0TEYN0SLHQB1EFR
Should we save this key for future use? [y/N] >>> y
4.7
2291.4921875
Wed, 23 Mar 2022 11:42:15
200 OK

```

```
Roshans-MBP:demo roshansoni$ cp ~/roshangit/iot/lesson3/system_info.py .  
Roshans-MBP:demo roshansoni$ cp ~/roshangit/iot/lesson7/rpi_spreadsheet.py
```

```
from system_info import get_temperature  
from oauth2client.service_account import ServiceAccountCredentials  
GDOCS_OAUTH_JSON = 'client_secret_1072685378926-uif7ne9hmrp731uhqfk9p5j74b701fjp.apps.googleusercontent.com'  
GDOCS_SPREADSHEET_NAME = 'rpidata'
```

```
[Roshans-MBP:demo roshansoni$ nano rpi_spreadsheet.py  
[Roshans-MBP:demo roshansoni$ python3 rpi_spreadsheet.p
```