	D	County	uv_avg
	2020-05-16	新竹縣	0
	2020-05-16	連江縣	0
	2020-05-16	花蓮縣	0
	2020-05-17	新北市	1.8333333333333333
	2020-05-17	桃園市	2.29
	2020-05-17	苗栗縣	2.3333333333333335
	2020-05-17	臺中市	1.9949999999999999
	2020-05-17	彰化縣	2
	2020-05-17	南投縣	1.97

```
import urllib.parse
import requests
import pymysql
import json
import time

def ADD_TABLE():
    con = pymysql.connect(host = 'localhost',user = 'root',passwd = 'n2
26165528',db = 'db_demo')
    cursor = con.cursor()

    # cursor.execute("DROP TABLE IF EXISTS db_uv") # 習慣性
# sql = """CREATE TABLE db_uv (
#         `County` VARCHAR(20),
#         `PublishAgency` VARCHAR(100),
#         `PublishTime` VARCHAR(200),
#         `SiteName` varchar(20),
#         `UVI` VARCHAR(20),
#         `WGS84Lat` varchar(20),
#         `WGS84Lon` varchar(20)
#         )
#         """
# cursor.execute(sql) # 根據需要建立一個表格
    cursor.execute("SELECT VERSION()")
    data = cursor.fetchone()
    print("Database version : %s " % data) # 結果表明已經連線成功
    for i in json_obj:
        print("County:", i["County"])
        print("PA:", i["PublishAgency"])
        print("PT:", i["PublishTime"])
```

```
        print("PT:", i["SiteName"])
        print("UVI:", i["UVI"])
        print("PT:", i["WGS84Lat"])
        print("PT:", i["WGS84Lon"])
        print('---')
        cursor.execute("INSERT INTO db_uv (County, PublishAgency, Publi
shTime, SiteName, UVI, WGS84Lat, WGS84Lon) VALUES (%s, %s, %s, %s, %s,
%s, %s)", (i["County"], i["PublishAgency"], i["PublishTime"], i["SiteNa
me"], i["UVI"], i["WGS84Lat"], i["WGS84Lon"]))
    cursor.execute("CREATE TABLE reg_uv AS (SELECT DISTINCT * FROM db_u
v)")#刪除重複資料
    cursor.execute("DELETE FROM db_uv")
    cursor.execute("INSERT INTO db_uv (SELECT * FROM reg_uv)")
    cursor.execute("DROP TABLE reg_uv")
    cursor.execute("CREATE or replace VIEW uv_view (D, County, uv_avg)
AS SELECT substring(PublishTime, 1, 10), County, AVG(UVI) FROM db_uv gr
oup by substring(PublishTime, 1, 10), County")
    con.commit()
    con.close()
headers = { 'User-Agent' : 'User-Agent:Mozilla/5.0' }
url = 'http://opendata.epa.gov.tw/webapi/Data/UV/?$orderby=PublishTime%
20desc&$skip=0&$top=1000&format=json'
data1 = urllib.request.Request(url , headers = headers )
response = urllib.request.urlopen(data1).read()
# json_obj = str(response, 'utf-8')
json_obj = json.loads(response.decode('utf-8'))


def saver():
    number = json_obj
    file_name = 'C:\\Users\\as722\\Desktop\\資料庫作業 0517\\test.py' #通
过扩展名指定文件存储的数据为 json 格式
    with open(file_name,'w') as file_object:
        json.dump(number,file_object)

def timer(n):
    while True:
        saver()
        ADD_TABLE()
```

```
        time.sleep(n)
# main
if __name__ == "__main__":

    timer(3600)
```

關鍵為

```
CREATE or replace VIEW uv_view (D, County, uv_avg) AS SELECT substring(
PublishTime, 1, 10), County, AVG(UVI) FROM db_uv group by substring(Pub
lishTime, 1, 10), County
```

這句 SQL 指令，而我截取前十位作為日期

也在過程中因為利用 urllib 發起的請求，UA 默認是 Python-urllib/3.5 而在 chrome 中訪問 UA 則是 User-Agent:Mozilla/5.0，因為服務器根據 UA 來判斷拒絕了 python 爬蟲。

所以在讀取資料之前加上

```
headers = { 'User-Agent' : 'User-Agent:Mozilla/5.0' }
url = 'http://opendata.epa.gov.tw/webapi/Data/UV/?$orderby=PublishTime%
20desc&$skip=0&$top=1000&format=json'
data1 = urllib.request.Request(url , headers = headers )
```

使的能夠成功讀取 JSON 資料