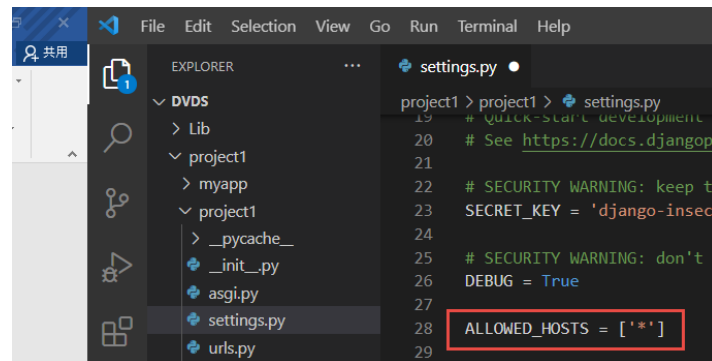


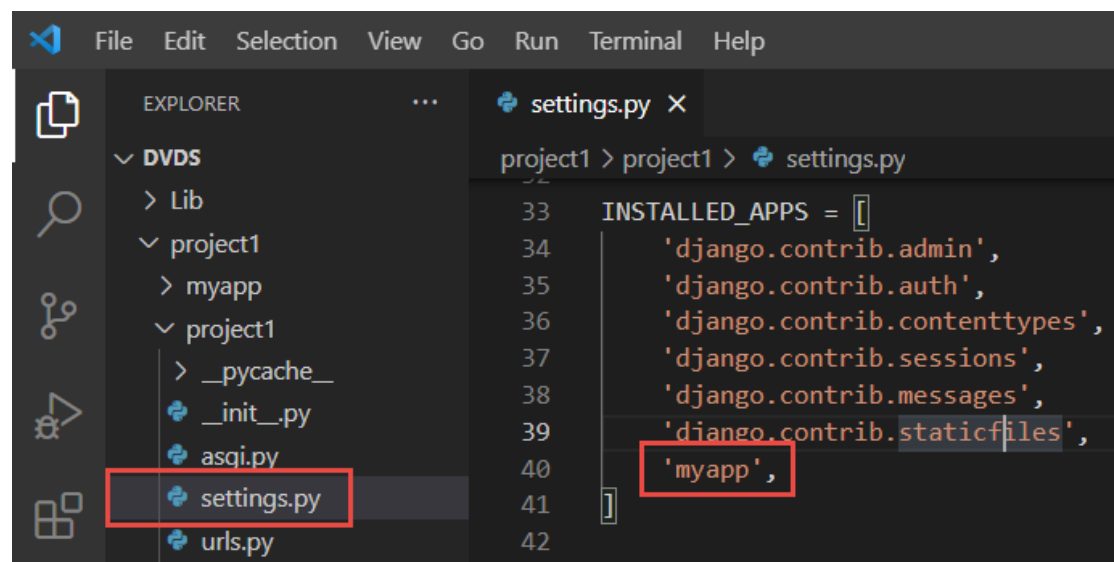
3-3 將溫度(DS18B20)傳到網路(以 django 為例 by windows)

➤ 雲端(以 django 為例 by windows)

修改 settings.py

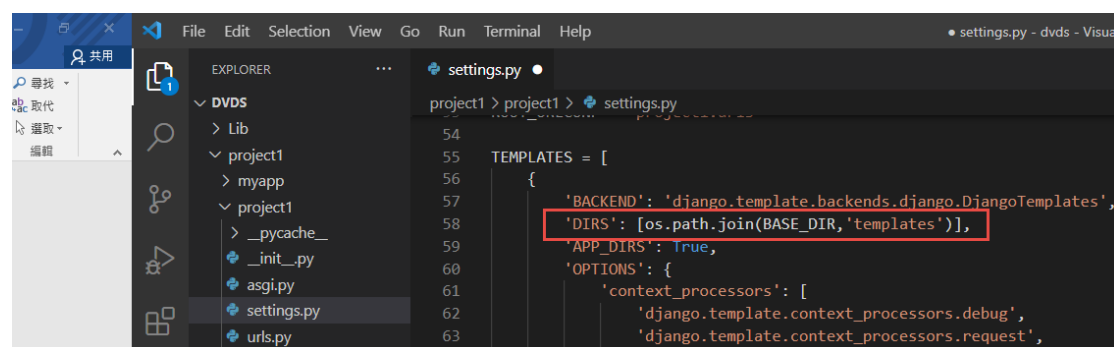


A screenshot of the Visual Studio Code editor. The Explorer pane on the left shows a project structure with folders 'DVDS', 'Lib', 'project1', and 'myapp'. Inside 'project1', there are files like 'urls.py', 'settings.py', and 'asgi.py'. The 'settings.py' file is selected and open in the main editor. The code in 'settings.py' shows various Django settings. The line `ALLOWED_HOSTS = ['*']` is highlighted with a red rectangle.

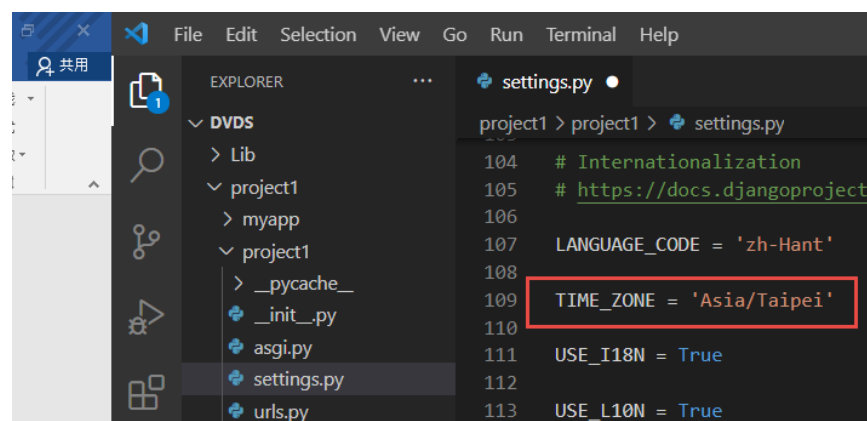


A screenshot of the Visual Studio Code editor. The Explorer pane on the left shows the same project structure. The 'settings.py' file is selected and open in the main editor. The code in 'settings.py' shows the `INSTALLED_APPS` list. The line `'myapp',` is highlighted with a red rectangle.

最前面要加「import os」



A screenshot of the Visual Studio Code editor. The Explorer pane on the left shows the same project structure. The 'settings.py' file is selected and open in the main editor. The code in 'settings.py' shows the `TEMPLATES` list. The line `'DIRS': [os.path.join(BASE_DIR, 'templates')],` is highlighted with a red rectangle.

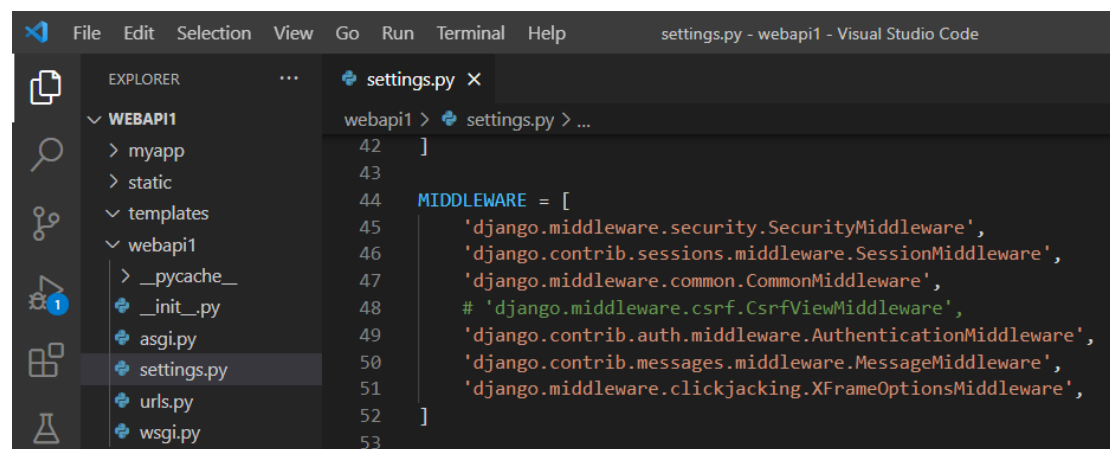


在使用 post 時會報錯 (Django 解決 403 forbidden 報錯)

第一種處理方式(全部禁止使用 csrf token):

Django 把項目中 settings.py 中 MIDDLEWARE 中的

'django.middleware.csrf.CsrfViewMiddleware' 註釋掉就行



第二種處理方式(禁止單一 url 使用 csrf token):

How to disable Django's CSRF validation?

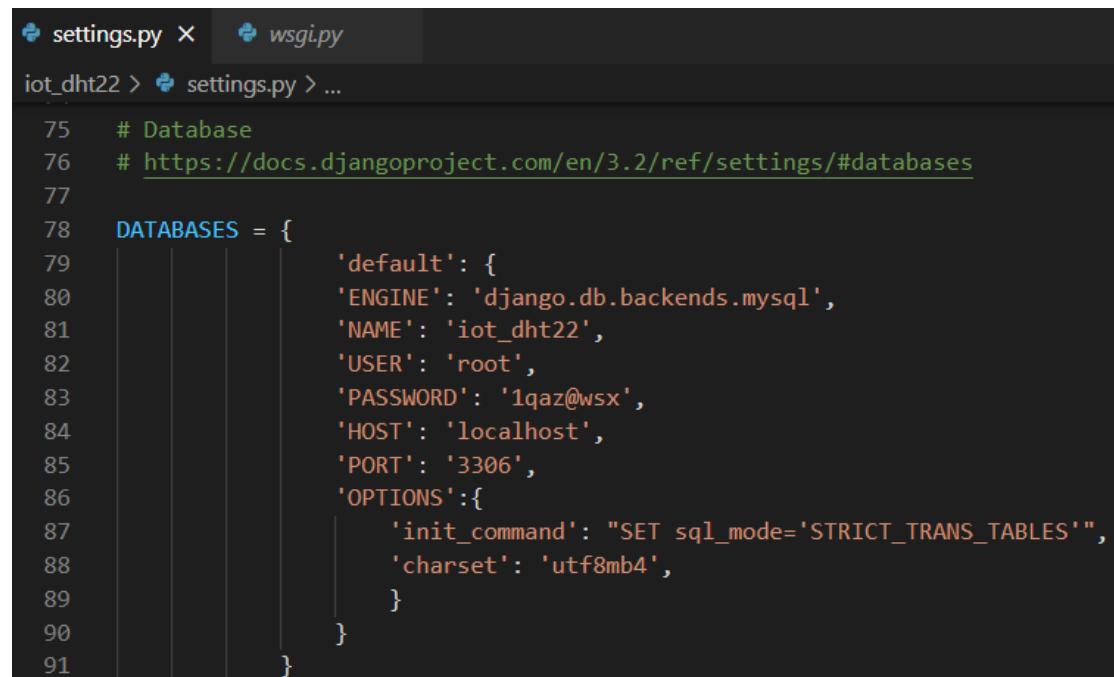
If you just need some views not to use CSRF, you can use `@csrf_exempt`:

```
from django.views.decorators.csrf import csrf_exempt

@csrf_exempt
def my_view(request):
    return HttpResponse('Hello world')
```

Reference:

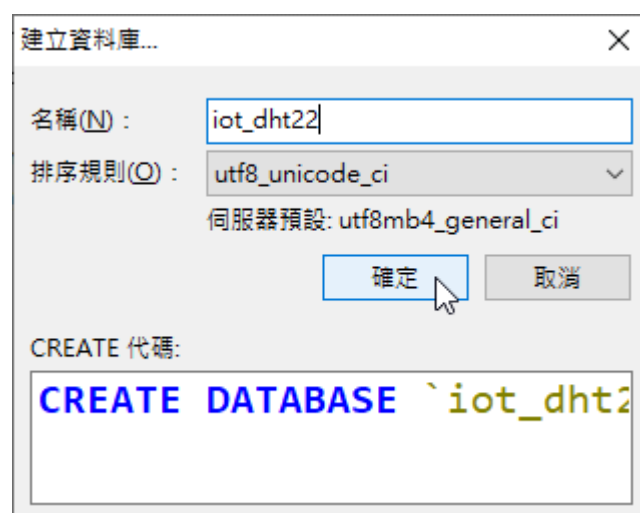
<https://stackoverflow.com/questions/16458166/how-to-disable-django-csrf-validation>



```
settings.py X wsgi.py
iot_dht22 > settings.py > ...

75 # Database
76 # https://docs.djangoproject.com/en/3.2/ref/settings/#databases
77
78 DATABASES = {
79     'default': {
80         'ENGINE': 'django.db.backends.mysql',
81         'NAME': 'iot_dht22',
82         'USER': 'root',
83         'PASSWORD': '1qaz@wsx',
84         'HOST': 'localhost',
85         'PORT': '3306',
86         'OPTIONS': {
87             'init_command': "SET sql_mode='STRICT_TRANS_TABLES'",
88             'charset': 'utf8mb4',
89         }
90     }
91 }
```

建立空的資料庫「iot_dht22」



```
# python manage.py makemigrations myapp (建立 migration 資料檔)
```

```
# python manage.py migrate (模型與資料庫同步)
```

```
(dvds) c:\dvds\iot_dht22>python manage.py makemigrations myapp
No changes detected in app 'myapp'

(dvds) c:\dvds\iot_dht22>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying sessions.0001_initial... OK
```

```
# dir
```

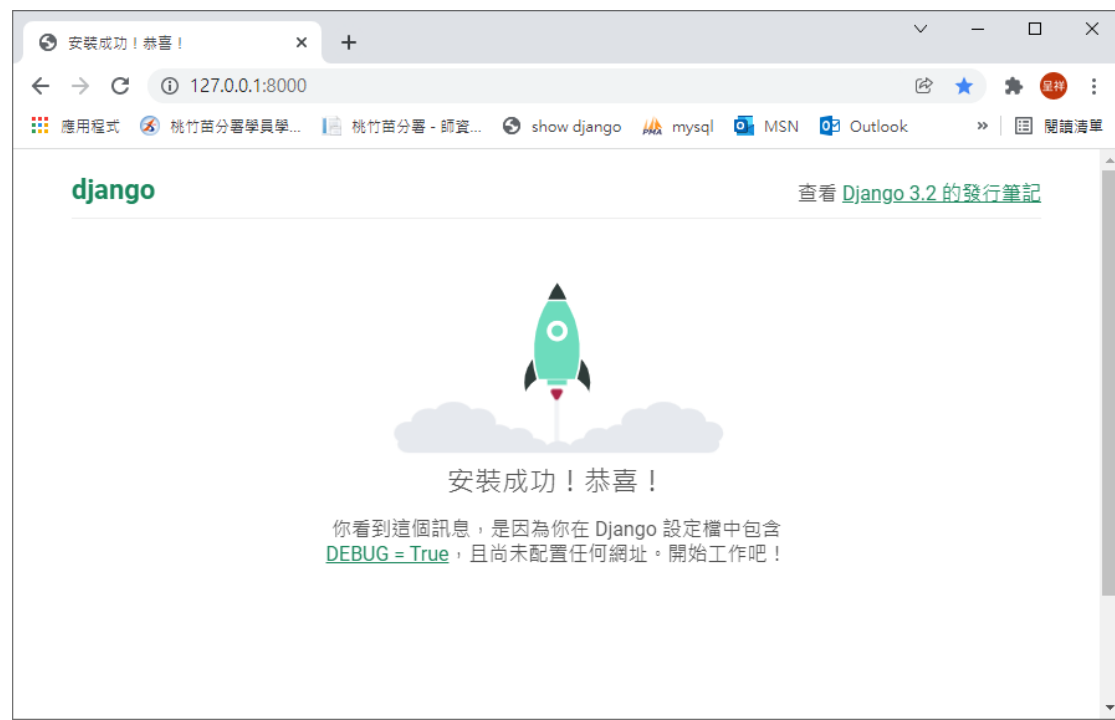
```
(dvds) c:\dvds\iot_dht22>dir
磁碟區 C 中的磁碟沒有標籤。
磁碟區序號: 8657-D998

c:\dvds\iot_dht22 的目錄
2021/12/16 下午 02:51 <DIR> .
2021/12/16 下午 02:51 <DIR> ..
2021/12/16 下午 02:51 <DIR> iot_dht22
2021/12/16 下午 02:50 687 manage.py
2021/12/16 下午 02:55 <DIR> myapp
2021/12/16 下午 02:51 <DIR> static
2021/12/16 下午 02:51 <DIR> templates
                1 個檔案 687 位元組
                6 個目錄 94,051,020,800 位元組可用
```

```
#python manage.py runserver 0.0.0.0:8000
```

```
(dvds) c:\dvds\iot_dht22>python manage.py runserver 0.0.0.0:8000
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
December 16, 2021 - 14:59:19
Django version 3.2.4, using settings 'iot_dht22.settings'
Starting development server at http://0.0.0.0:8000/
Quit the server with CTRL-BREAK.
```

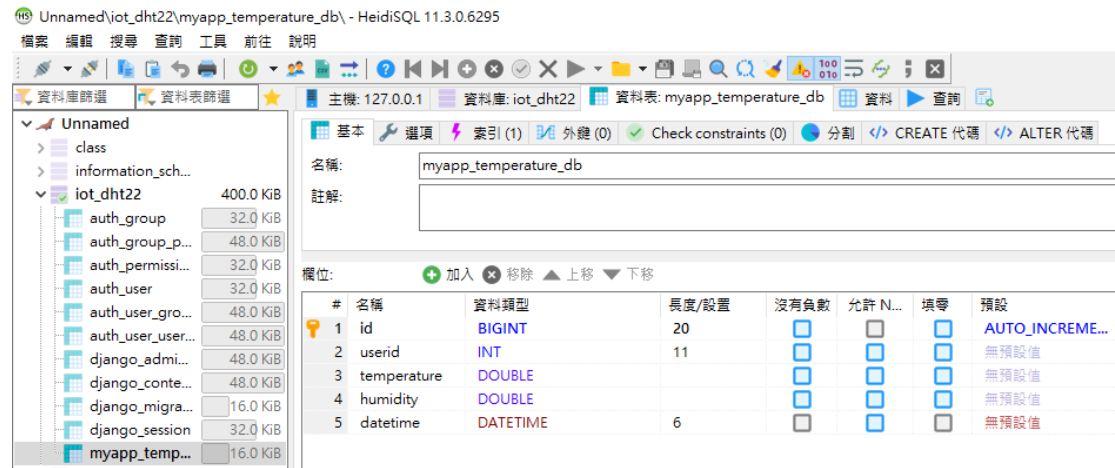


➤ 建立資料庫

```
# vim myapp/models.py
```

```
settings.py  models.py X
myapp > models.py > temperature_db
1  from django.db import models
2  import django.utils.timezone as timezone
3
4  class temperature_db(models.Model):
5      |   userid = models.IntegerField(null=False)
6      |   temperature = models.FloatField(null=False)
7      |   humidity = models.FloatField(null=False)
8      |   datetime = models.DateTimeField(default = timezone.now)
9
```

```
# python manage.py makemigrations myapp (建立 migration 資料檔)
# python manage.py migrate (模型與資料庫同步)
```



➤ 撰寫 django 程式

新增瀏覽位置(urls.py):

```
# sudo vim iot_dht22/urls.py
```

```
16 from django.contrib import admin
17 from django.urls import path
18 from myapp import views
19
20 urlpatterns = [
21     path('admin/', admin.site.urls),
22     path('show_history_temperature/', views.show_history_temperature),
23     path('add_temperature/<str:mode>/', views.add_temperature),
24     path('api_add_temperature/', views.api_add_temperature),
25 ]
26 ]
```

新增 view functions

```
# sudo vim myapp/views.py
```

```

1  from django.shortcuts import render
2  from django.http import HttpResponse
3  from django.db import connections
4  from datetime import datetime
5  from django.shortcuts import redirect
6
7  def show_history_temperature(request):
8      sql = "SELECT * FROM myapp_temperature_db ORDER BY DATETIME DESC"
9      cursor = connections['default'].cursor() #連接資料庫
10     cursor.execute(sql,[]) #執行sql語法
11     result = cursor.fetchall() #取得資料
12     # print(result)
13
14     #轉換格式
15     field_name=cursor.description #取得資料表欄位名稱
16     # print(field_name)
17
18     resultList=[]
19     for data in result:
20         # print(data)
21         i=0
22         dict_data={}
23         while i< len(field_name):
24             dict_data[field_name[i][0]]=data[i] #dict_data[field_name[0][0]]
25             i = i + 1
26         # print(dict_data)
27         dict_data["datetime"]=str(dict_data["datetime"]) #將時間轉為字串
28         resultList.append(dict_data)
29     print(resultList)
30     # return HttpResponse("test...")
31     return render(request,"show_history_temperature.html",locals())

```

```

33 def add_temperature(request, mode=None):
34     if mode == "edit":
35         sensor_id = request.GET["sensor_id"]
36         temperature = request.GET["temperature"]
37         humidity = request.GET["humidity"]
38         s_datetime = request.GET["s_datetime"]
39         #可檢查版本
40         sql = "INSERT INTO myapp_temperature_db (sensor_id,temperature,humidity,datetime)"
41         sql += "VALUES('%s','%s','%s','%s')"

```

```

53 # 1. Import the csrf_exempt decorator
54 from django.views.decorators.csrf import csrf_exempt
55 # 2. Exempt the view from CSRF checks
56 @csrf_exempt
57 def api_add_temperature(request):
58     try:
59         if request.method == "GET":
60             sensor_id = request.GET["sensor_id"]
61             temperature = request.GET["temperature"]
62             humidity = request.GET["humidity"]
63         elif request.method == "POST":
64             sensor_id = request.POST["sensor_id"]
65             temperature = request.POST["temperature"]
66             humidity = request.POST["humidity"]
67     except:
68         return HttpResponse("add error")
69     try:
70         s_datetime = datetime.now()
71         s_datetime = s_datetime.strftime("%Y/%m/%d %H:%M:%S") #格式化
72         print(s_datetime)
73         #可檢查版本
74         sql = "INSERT INTO myapp_temperature_db (sensor_id,temperature,humidity,datetime)"
75         sql += "VALUES('%s','%s','%s','%s')"
76         sql %= (sensor_id,temperature,humidity,s_datetime)
77         # print(sql)
78         cursor = connections["default"].cursor()
79         cursor.execute(sql,[])
80         cursor.close()
81     except:
82         return HttpResponse("sql execute error")
83     return HttpResponse("新增成功!")

```

測試結果：

```
# sudo python3 manage.py runserver 0.0.0.0:8000
```

POST 192.168.57.246:8000/api_add_temperature/

Save

POST 192.168.57.246:8000/api_add_temperature/ Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

KEY	VALUE	DESCRIPTION	Bulk Edit
<input checked="" type="checkbox"/> sensor_id	9		
<input checked="" type="checkbox"/> temperature	40		
<input checked="" type="checkbox"/> humidity	90		
Key	Value	Description	

Body Cookies Headers (7) Test Results 200 OK 23 ms 253 B Save Response

Pretty Raw Preview Visualize HTML

1 新增成功!

```
# sudo vim templates/show_history_temperature.html
```



```

show_history_temperature.html X
templates > show_history_temperature.html
1 |!DOCTYPE html|
2 |<html lang="en">
3 |<head>
4 |   <meta charset="UTF-8">
5 |   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6 |   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7 |   <title>Document</title>
8 |   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-gH2yIqKdNHPEq0n4Mqqa/
HKGItHskIHeL5AyhkyV8i59U5AR6csBvApHhN1/v118x" crossorigin="anonymous">
9 |   <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.5/dist/umd/popper.min.js" integrity="sha384-Xe+8cL9oJa6tN/veChSP7q
+mnSPaj5Bcu9mPX5F5xIGe00VittagI5lorf0EI7Vk" crossorigin="anonymous"></script>
10 |   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/js/bootstrap.min.js" integrity="sha384-ODmDlVzN+pFdexxHEHFBQH3/9/
vQ9uori45z4JjnfSRydbmQbml5titQ0cul0zyK" crossorigin="anonymous"></script>
11 | </head>
12 | <body>
13 |   <!-- <h1>test</h1> -->
14 |   <h3><a href="/add_temperature/load/">新增溫度</a></h3>
15 |   <table class="table caption-top">
16 |     <caption>歷史溫度顯示</caption>
17 |     <thead>
18 |       <tr>
19 |         <th scope="col">sensor編號</th>
20 |         <th scope="col">溫度</th>
21 |         <th scope="col">濕度</th>
22 |         <th scope="col">時間</th>
23 |       </tr>
24 |     </thead>
25 |     <tbody>
26 |       {% for data in resultList %}
27 |       <tr>
28 |         <th scope="row">{{ data.sensor_id }}</th>
29 |         <td>{{ data.temperature }}</td>
30 |         <td>{{ data.humidity }}</td>
31 |         <td>{{ data.datetime }}</td>
32 |       </tr>
33 |       {% endfor %}
34 |     </tbody>
35 |   </table>
36 | </body>
37 | </html>

```

```
# sudo vim templates/add_temperature.html
```

```
<> add_temperature.html X
templates > <> add_temperature.html
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta http-equiv="X-UA-Compatible" content="IE=edge">
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8      <title>學生資料管理系統-修改資料一(GET)</title>
9      <style>
10         h1,
11         h3 {
12             text-align: center;
13         }
14
15         table {
16             margin-left: auto;
17             margin-right: auto;
18         }
19
20         table,
21         th,
22         td {
23             border: 1px solid black;
24         }
25     </style>
26 </head>
```

```

28 <body>
29   <h1>新增溫濕度頁面(GET)</h1>
30   <a href="/show_history_temperature/">
31     <h3>溫濕度歷史資訊</h3>
32   </a>
33   <form action="/add_temperature/edit/" method="get">
34     <table>
35       <tr>
36         <th>Sensor_ID</th>
37         <td><input type="text" id="sensor_id" name="sensor_id"></td>
38       </tr>
39       <tr>
40         <th>溫度</th>
41         <td><input type="text" id="temperature" name="temperature"></td>
42       </tr>
43       <tr>
44         <th>濕度</th>
45         <td><input type="text" id="humidity" name="humidity"></td>
46       </tr>
47       <tr>
48         <th>日期</th>
49         <td><input type="datetime-local" id="s_datetime" name="s_datetime"></td>
50       </tr>
51       <tr>
52         <th colspan="2" style="text-align:center;">
53           <input type="submit" name="button" id="button" value="儲存">
54           <input type="reset" name="button2" id="button2" value="重設">
55         </th>
56       </tr>
57     </table>
58   </form>
59 </body>
60 </html>

```

測試結果：

```
# sudo python3 manage.py runserver 0.0.0.0:8000
```

Document x +

127.0.0.1:8000/show_history_temperature/

桃竹苗分署學員學... 桃竹苗分署 - 師資... mysql MSN Outlook Gmail Google 翻譯 Yahoo 奇摩 YouTube tony Google 翻譯

[新增溫度](#)

歷史溫濕度顯示

sensor編號	溫度	濕度	時間
33	33.0	33.0	2022-09-30 15:27:00
3	100.0	102.0	2022-09-30 15:22:00
3	40.25	80.25	2022-09-26 15:05:00
33	33.0	33.0	2022-09-20 15:04:00
2	2.0	2.0	2022-09-15 14:15:00
2	2.0	2.0	2022-09-15 14:15:00
2	2.0	2.0	2022-09-15 14:15:00
3	28.06	80.2	2022-09-12 10:11:03



參考網址:

<https://docs.python.org/zh-tw/3/howto/urllib2.html>

➤ 查尋 python 路徑

```
# which python
```

➤ 加入 crontab(排程)

```
# crontab -e
```

```
24 */1 * * * * python /home/pi/python_code/dht22_read.py
25 #55 10 * * * sudo python /home/pi/python_code/dht22_read_repeat.py
26
```

補充:

若選擇錯誤的編輯器，請在家目錄刪除「.selected_editor」

```
# service cron status
```

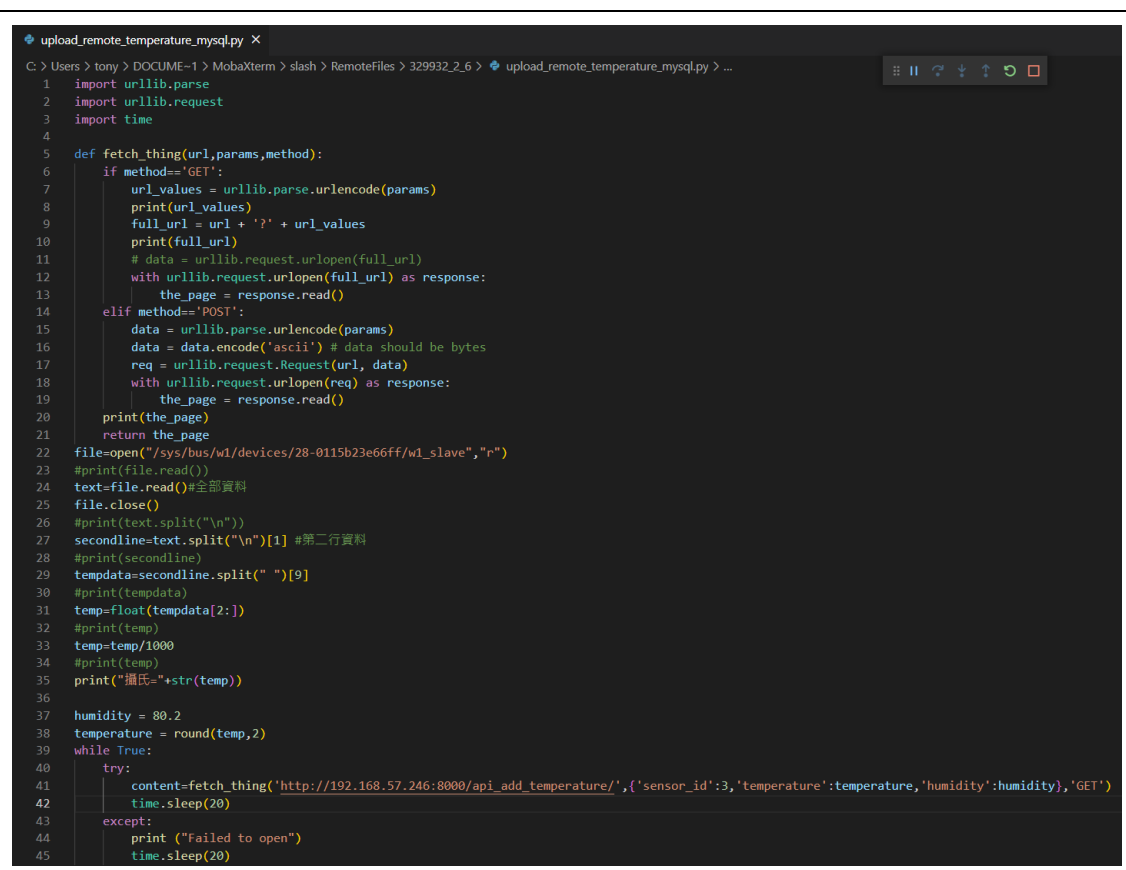
(查看排程狀態)

```
# ps -aux | grep dht22 (列出背景行程)
```

```
# kill 9 number
```

(關閉某個執行中的行程)

- 將 raspberry 內的溫度傳至雲端(使用 GET or POST)，利用開機時自動執行，且 20 秒上傳一次



```

upload_remote_temperature_mysql.py
C:\Users> tony > DOCUME-1 > MobaXterm > slash > RemoteFiles > 329932_2_6 > upload_remote_temperature_mysql.py > ...

1  import urllib.parse
2  import urllib.request
3  import time
4
5  def fetch_thing(url,params,method):
6      if method=='GET':
7          url_values = urllib.parse.urlencode(params)
8          print(url_values)
9          full_url = url + '?' + url_values
10         print(full_url)
11         # data = urllib.request.urlopen(full_url)
12         with urllib.request.urlopen(full_url) as response:
13             the_page = response.read()
14         elif method=='POST':
15             data = urllib.parse.urlencode(params)
16             data = data.encode('ascii') # data should be bytes
17             req = urllib.request.Request(url, data)
18             with urllib.request.urlopen(req) as response:
19                 the_page = response.read()
20         print(the_page)
21         return the_page
22     file=open("/sys/bus/w1/devices/28-0115b23e66ff/w1_slave", "r")
23     #print(file.read())
24     text=file.read()#全部資料
25     file.close()
26     #print(text.split("\n"))
27     secondline=text.split("\n")[1] #第二行資料
28     #print(secondline)
29     tempdata=secondline.split(" ")[9]
30     #print(tempdata)
31     temp=float(tempdata[2:])
32     #print(temp)
33     temp=temp/1000
34     #print(temp)
35     print("攝氏="+str(temp))
36
37     humidity = 80.2
38     temperature = round(temp,2)
39     while True:
40         try:
41             content=fetch_thing('http://192.168.57.246:8000/api_add_temperature/',{'sensor_id':3,'temperature':temperature,'humidity':humidity},'GET')
42             time.sleep(20)
43         except:
44             print ("Failed to open")
45             time.sleep(20)

```

upload_remote_temperature_mysql.py

```
# crontab -e
```

```
@reboot /usr/bin/python /home/pi/python_code/dht22_url_save_repeat.py &
```

其中&為背景執行

```
# service cron status
```

(查看排程狀態)

```
pi@raspberrypi:~ $ service cron status
● cron.service - Regular background program processing daemon
   Loaded: loaded (/lib/systemd/system/cron.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2021-07-02 15:52:05 CST; 26min ago
     Docs: man:cron(8)
   Main PID: 311 (cron)
   CGroup: /system.slice/cron.service
           └─311 /usr/sbin/cron -f
             └─338 /usr/bin/python /home/pi/python_code/dht22_url_save_repeat.py
```

```
# ps -aux | grep dht22 (列出背景行程)
```

```
pi@raspberrypi:~ $ ps -aux | grep dht22
pi      338  0.0  0.9 13264  9016 ?        S    15:52   0:00 /usr/bin/python /home/pi/python_code/dht22_url_save_repeat.py
pi      1245  0.0  0.0   4372   556 pts/1    S+   16:19   0:00 grep --color=auto dht22
```