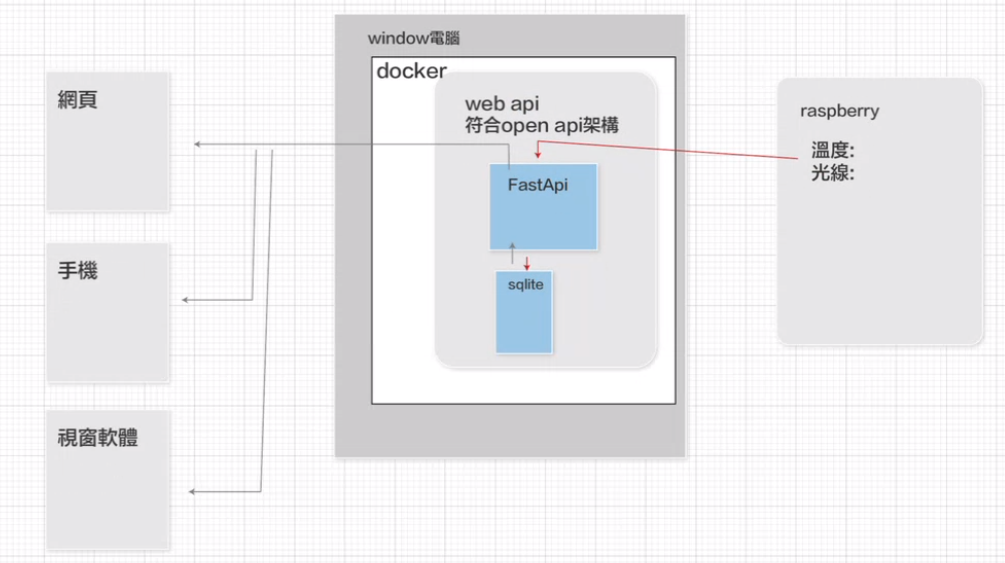
用Docker在Windows架構Linux環境



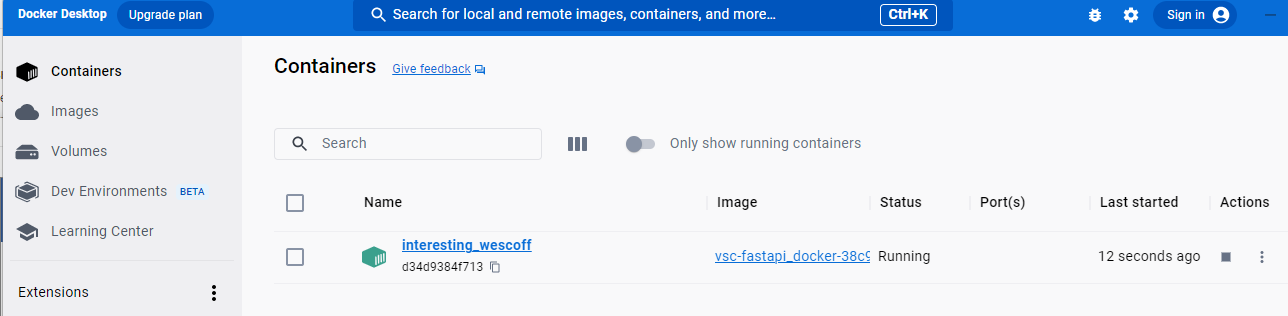
政府資料標準平台

<https://data.gov.tw/>

使用API取用平台

<https://schema.gov.tw/oas.html>





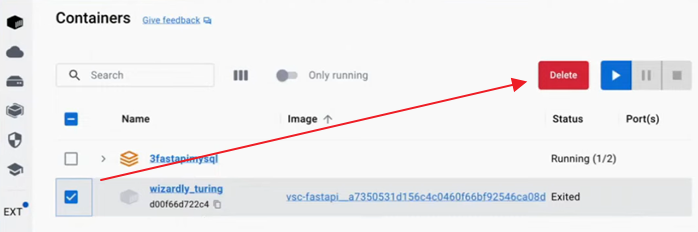


Docker Images 刪除步驟

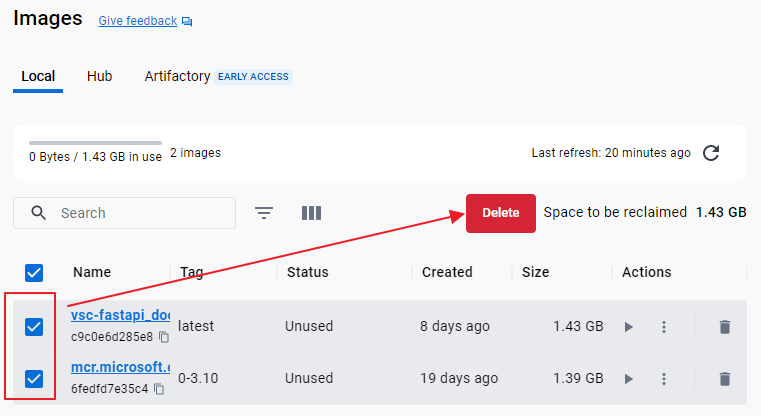
1. 刪除專案資料夾,

D:/FASTAPI\_DOCKER

1. github上的專案也刪除
2. 刪DOCKER Containers



4.刪Docker裡的 Images



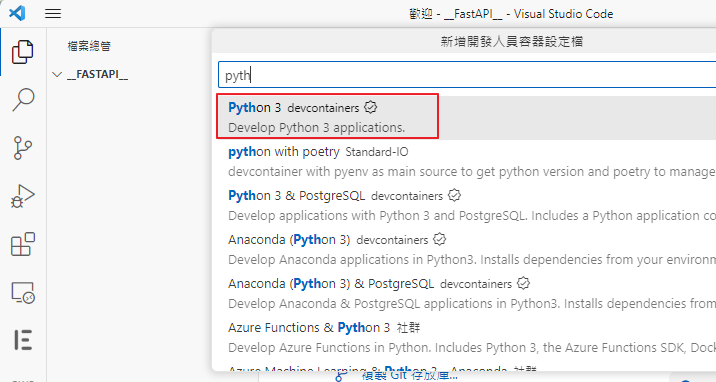
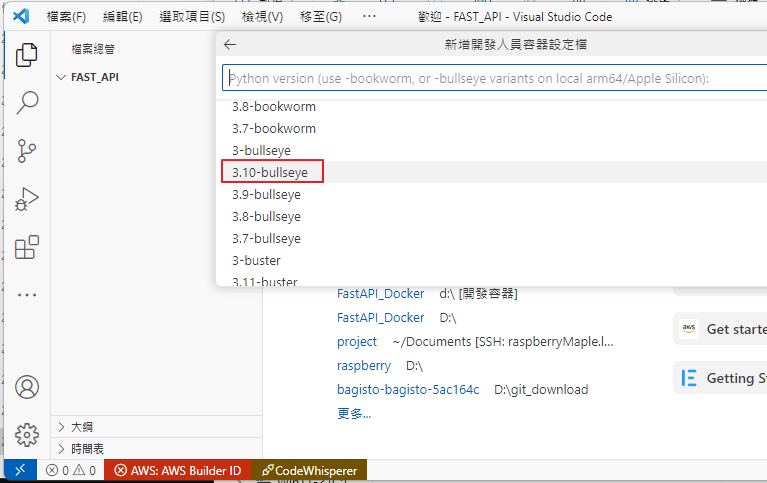
重要觀念：

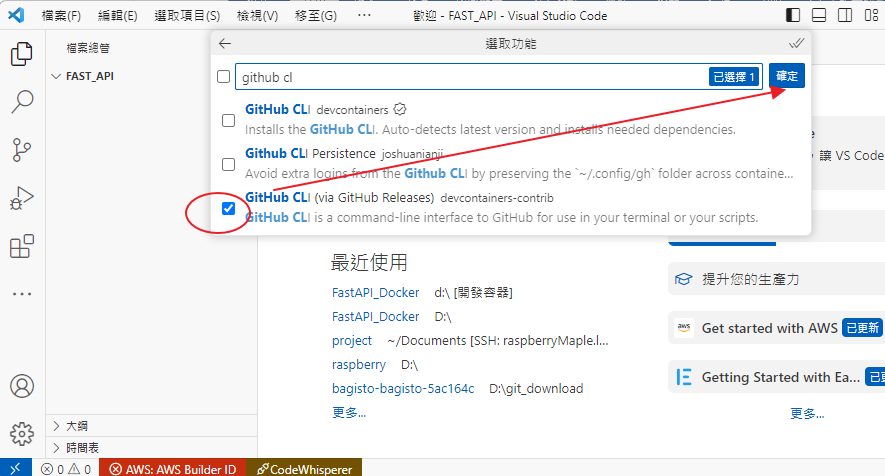
devcontainer .json檔

新建docker 電腦環境步驟

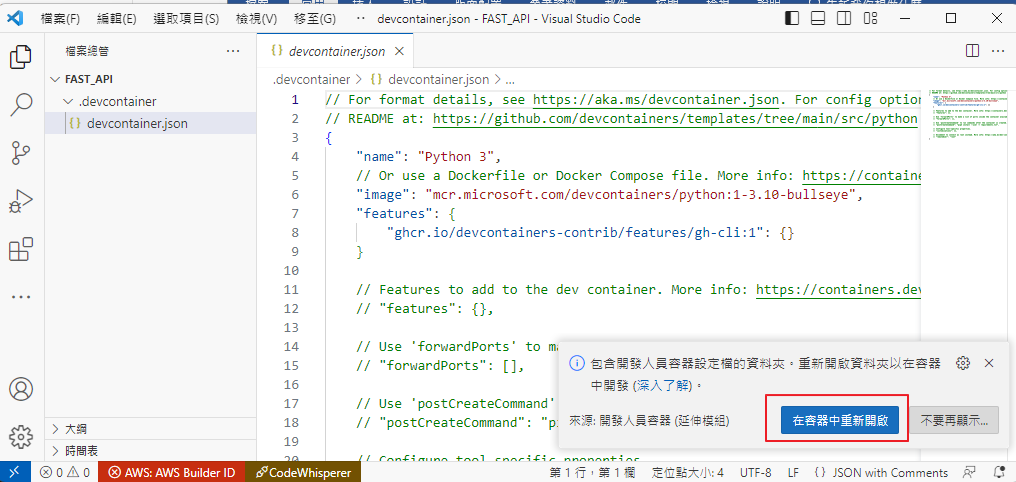
1. 先建立專案資料夾 D:/FAST\_API
2. 進入vsc 選資夾後, 再點選左下角開始設定
3. 新增開發人員設定檔



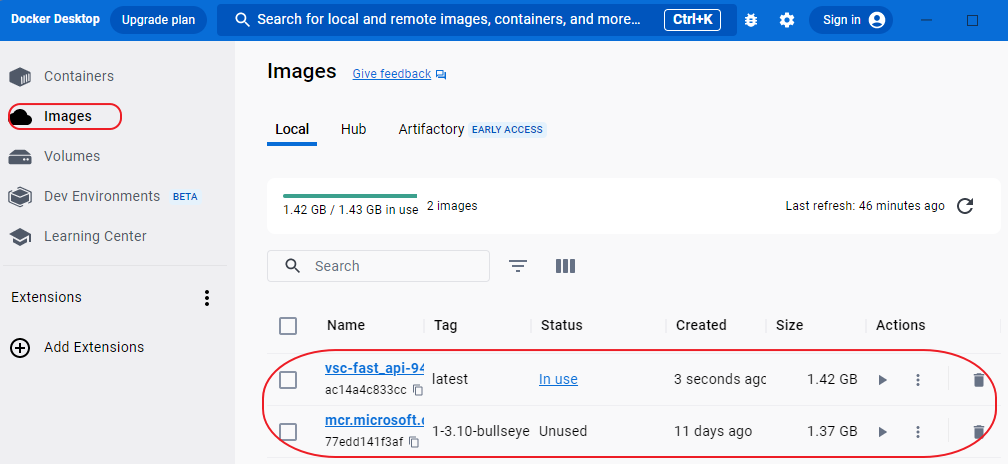
1. 選Python 3 devcontainers
2. 選3.10-bullseys
3. 選GitHub CLI (via…) 工具 ,然後按確定



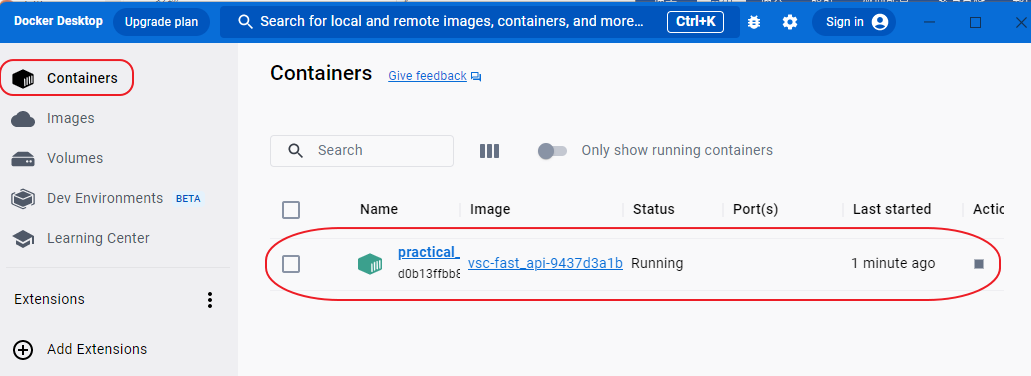
最後按 在容器中重新開啟, 等待安裝完成



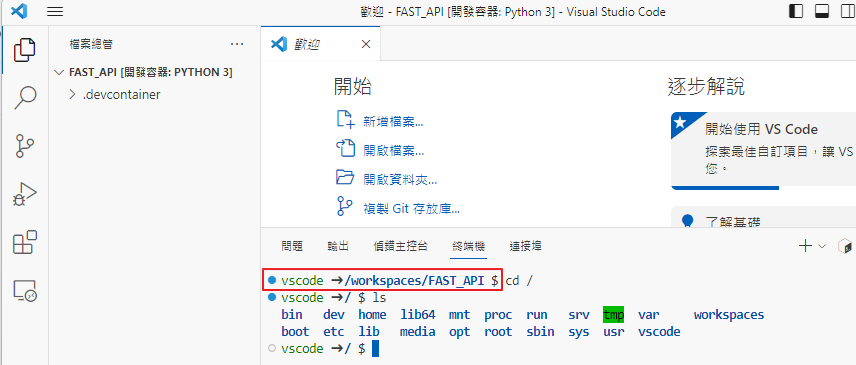
在 Docker Desktop Images 自動生成二個檔



在Containers建立一個檔案, 並正在 Running 中

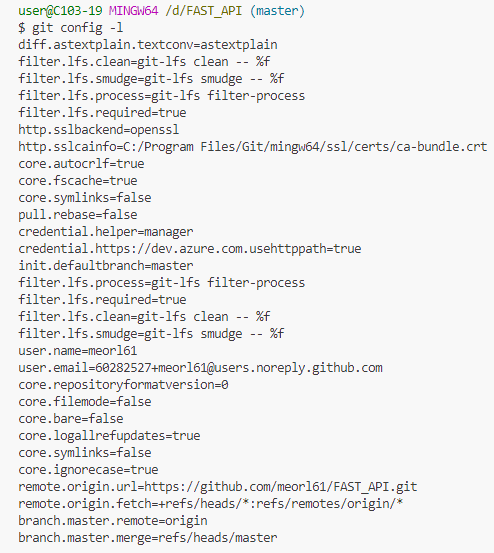


vsc 顯示

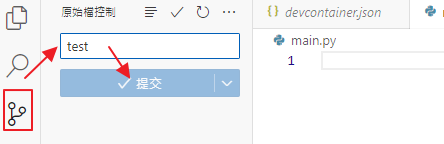


開始設定 git

1. 查詢 git –version
2. git config -l



1. git init 加入 git
2. 選取 commit



1. 發布, 會提示建立遠端資料夾

Python

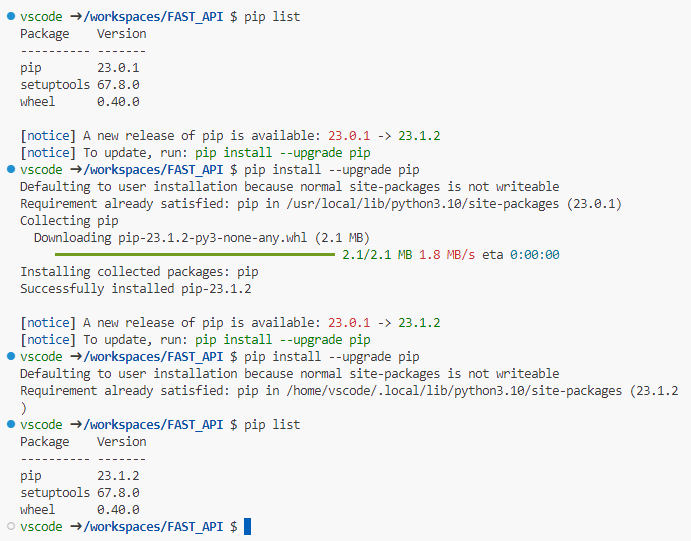
使用內部module , 用 **import (例如： import math**

**使用外部module, 用 fastApi**



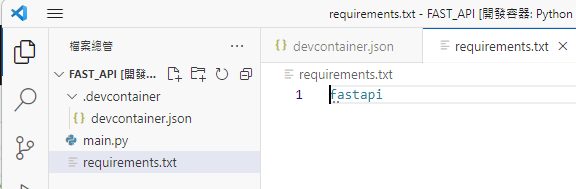
**pip 套件管理程式 (管理外部 module)**

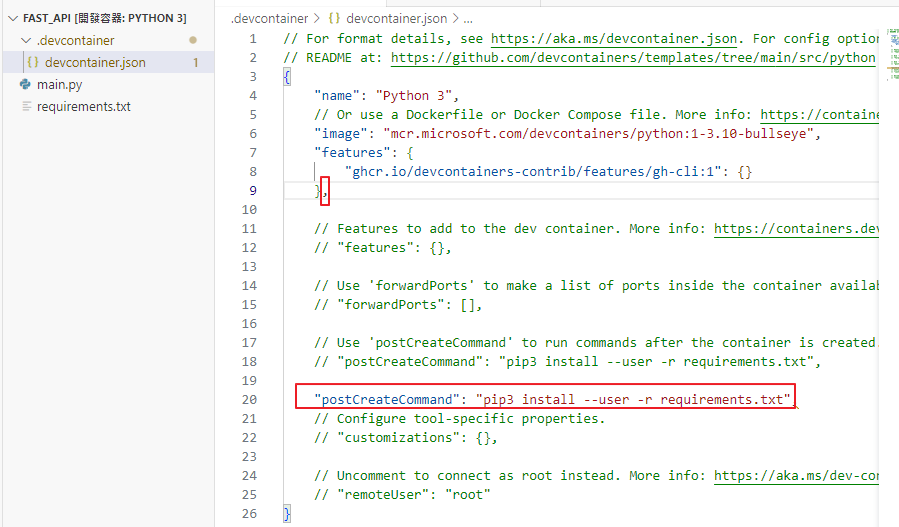
**pip list**



**安裝Fastapi**

[**https://fastapi.tiangolo.com/zh/**](https://fastapi.tiangolo.com/zh/)

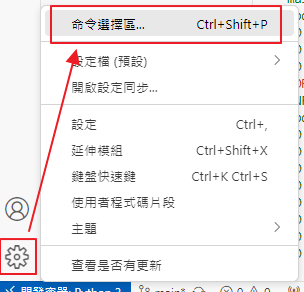
1. **每個容器安裝 https://pypi.org/project/fastapi/**
2. **安裝一次, 自動安裝**
3. **建立安裝檔 requirements.txt (固定名稱不可改)**
4. **輸入fastapi**
5. **修改 devcontainer.json**



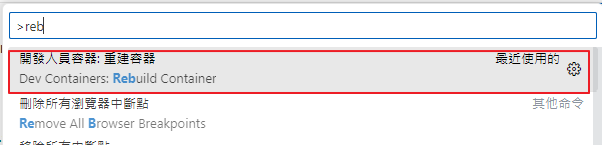
再加一個moudle: uvicorn



按F1, 或按左下角齒輪->命令選擇區



輸入 rebuild , 選 重建容器



以上, FastAPI 環境已設定完成

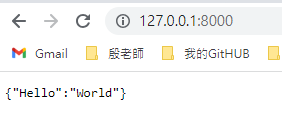
在 main.py 輸入程式



然後在終端機輸入 uvicorn main:app --reload



網頁要設定公開



---

將docker 使用github 的 codespace 做

Main.py

from fastapi import FastAPI

from datetime import datetime

app = FastAPI()

@app.get("/")

def read\_root():

return {"Hello": "rebort"}

@app.get("/items/{item\_id}")

async def read\_item(item\_id):

return {"item\_id": item\_id}

#query parameter

@app.get("/raspberry")

async def read\_items(time:datetime = datetime.now(),light:float = 0.0, temperature:float=0.0):

return {

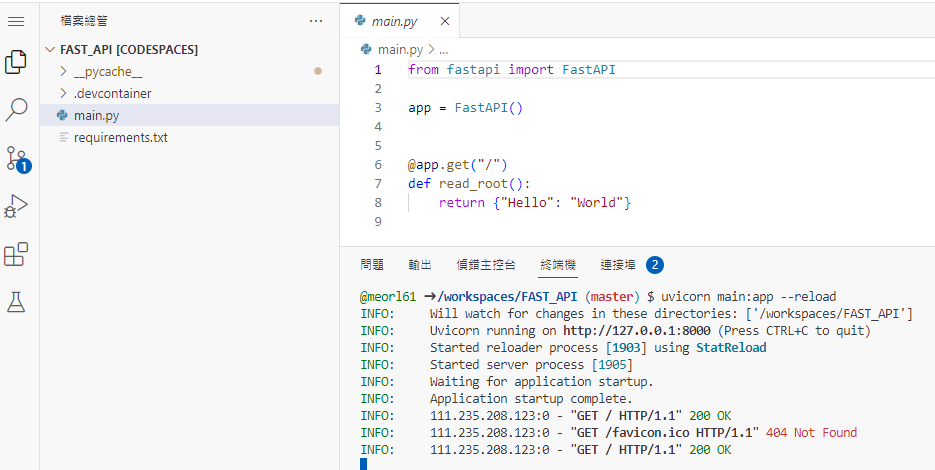
"時間":time.strftime("%Y%m%d %H:%M:%S"),

"光線":light,

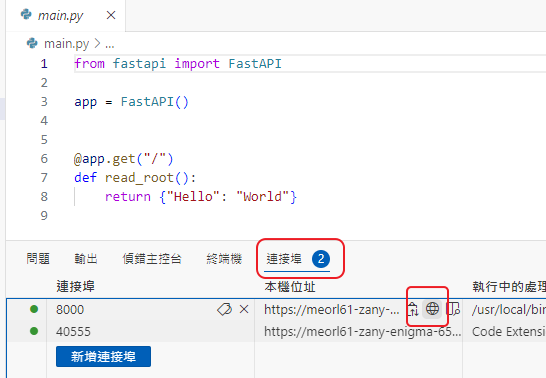
"溫度":temperature

}

然後在終端機輸入 uvicorn main:app --reload



在連接埠點選 8000 的網址



Copy 網址到raspberry執行



修改raspberry lesson6\_1.py程式

import gpiozero as zero

from time import sleep

import RPi.GPIO as GPIO

#連線遠端

import requests

if \_\_name\_\_ == "\_\_main\_\_":

mcp3008\_ch7 = zero.MCP3008(channel = 7)

mcp3008\_ch6 = zero.MCP3008(channel = 6)

try:

while True:

value\_ch7 = round(mcp3008\_ch7.value\*100)

value\_ch6 = round(mcp3008\_ch6.value\*100\*3.3\*2)

if value\_ch7 > 11:

print("光線亮")

else:

print("光線暗")

print("光敏電阻値: ", value\_ch7)

print("LM35",value\_ch6)

response=requests.get(f'https://meorl61-zany-enigma-6576x9469vjfr5j6-8000.preview.app.github.dev/raspberry?light={value\_ch7}&temperature={value\_ch6}')

if response.ok:

print("上傳成功")

print(response.text)

else:

print(response.status\_code)

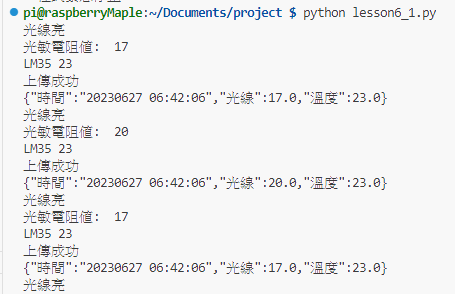
sleep(5)

except KeyboardInterrupt:

GPIO.cleanup()

print("程式緊急停止")

執行結果:



修改lesson6

import gpiozero as zero

from time import sleep

import RPi.GPIO as GPIO

from datetime import datetime

#連線遠端

import requests

if \_\_name\_\_ == "\_\_main\_\_":

mcp3008\_ch7 = zero.MCP3008(channel = 7)

mcp3008\_ch6 = zero.MCP3008(channel = 6)

try:

while True:

value\_ch7 = round(mcp3008\_ch7.value\*100)

value\_ch6 = round(mcp3008\_ch6.value\*100\*3.3\*2)

if value\_ch7 > 11:

print("光線亮")

else:

print("光線暗")

datetimeStr = datetime.now().strftime("%Y-%m-%d %H:%M:%S")

print("光敏電阻値: ", value\_ch7)

print("LM35",value\_ch6)

response=requests.get(f'https://meorl61-zany-enigma-6576x9469vjfr5j6-8000.preview.app.github.dev/raspberry?time={datetimeStr}&light={value\_ch7}&temperature={value\_ch6}')

if response.ok:

print("上傳成功")

print(response.text)

else:

print(response.status\_code)

sleep(5)

except KeyboardInterrupt:

GPIO.cleanup()

print("程式緊急停止")

------------------------------------------------------

<https://github.com/roberthsu2003/python-SQLite-MySQL>



修改 main.py

from fastapi import FastAPI

from datetime import datetime

import sqlite3

from sqlite3 import Error

app = FastAPI()

def create\_connection(db\_file):

conn = None

try:

conn = sqlite3.connect(db\_file)

except Error as e:

print(e)

return conn

def create\_table(conn):

sql\_projects = """

CREATE TABLE IF NOT EXISTS iot1(

id INTEGER PRIMARY KEY AUTOINCREMENT,

date TEXT NOT NULL,

light INTEGER NOT NULL,

temperature REAL NOT NULL

);

"""

try:

cursor = conn.cursor()

cursor.execute(sql\_projects)

except Error as e:

print(e)

def insert\_project(conn, project):

sql = """

INSERT INTO iot1(date,light,temperature)

VALUES(?,?,?)

"""

cursor = conn.cursor()

cursor.execute(sql,project)

conn.commit()

@app.get("/")

def read\_root():

return {"Hello": "rebort"}

@app.get("/items/{item\_id}")

async def read\_item(item\_id):

return {"item\_id": item\_id}

#query parameter

@app.get("/raspberry")

#async def read\_items(time:datetime = datetime.now(),light:float = 0.0, temperature:float=0.0):

async def read\_items(time:str = datetime.now().strftime("%Y%m%d %H:%M:%S"),light:float = 0.0, temperature:float=0.0):

#"時間":time.strftime("%Y%m%d %H:%M:%S"),

conn = create\_connection('data.db')

create\_table(conn)

insert\_project(conn, (time,light,temperature))

conn.close()

return {

"時間":time,

"光線":light,

"溫度":temperature

}