

# 氣象資料開放平臺的資料擷取

- 在中央氣象局的氣象資料開放平台網頁註冊免費的帳號來獲得授權碼
  - 劉奕的作法是直接深入解析資料底層
    - 直接處理網頁上的文字
    - 不需要另外註冊帳號就可以獲得中央氣象局的資料
  - 個人覺得透過註冊帳號後取得的資料來源比較穩定，而且不會因氣象局改變資料欄位位置而需要修改程式
- 程式概括
  - 1. 下載並解壓縮CWB官方檔案
  - 2. 轉換原始檔案並擷取資料 (利用xmltodict 第三方模組)
  - 3. 壓縮並備份資料
  - 4. 合併新舊資料 (使用強大的pandas 第三方模組)
- 每日新增並合併前次海象監測資料(csv) <http://60.249.144.60:8084>



# 氣象資料開放平臺的授權碼



🏠 > 會員資訊 > API授權碼

會員基本資料

API授權碼

## API授權碼

本平臺提供透過URL下載檔案以及 RESTful API 資料擷取方法取用資料，惟因本平臺採用會員服務機制，需帶入資料項目代碼以及有效會員之授權碼，方可取得各式開放資料。其中，資料項目代碼可至資料清單列表查詢。

### 一、取得授權碼

會員之授權碼可於下方按鈕取得

取得授權碼

### 二、更新授權碼

一旦更新授權碼後，舊的授權碼將永久失效，並且更新授權碼後七日內無法再進行更新。

# 1. 下載並解壓縮 CWB 官方檔案

```
# (1) download & decompress cwb mmc 30-day zip file
headers = {
    'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML
}

URL = 'https://opendata.cwb.gov.tw/fileapi/v1/opendataapi/O-B0075-002?Authorizat
    網址

res = requests.get(URL, headers=headers)

with open('./O-B0075-002.zip', mode='wb') as fp_w:
    for chunk in res.iter_content(100000):
        fp_w.write(chunk)
    下載 zip 格式的檔案

with zipfile.ZipFile('./O-B0075-002.zip') as myzip:
    myzip.extractall(FOLDER_XML)
    解壓縮 zip 檔案

print('1. download & decompress zip (complete)')
```

<https://opendata.cwb.gov.tw/fileapi/v1/opendataapi/O-B0075-002?Authorization=CWB-xxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx&downloadType=WEB&format=ZIP>

以上為開放資料的完整網址，記得 xxx-xxx-xxx-xxx-xxx 換成真正的授權碼

## 2. 轉換原始檔案並擷取資料

```
# (2) convert xml(decompressed files) into csv
```

```
files_path = FOLDER_XML + '*.xml'  
filenames_source = glob.glob(files_path)  
for filename_source in filenames_source:  
    filename_raw = os.path.basename(filename_source).split('.')[0]  
    filename_csv = f'{FOLDER_CSV}{filename_raw}.csv'
```

收集 xml 輸入檔案

命名 csv 輸出檔名

```
with open(filename_source, mode='rb') as f:  
    # get a list of 30-day records & every record is a dict with hierarchy  
    my_dict = xmltodict.parse(f)  
    a = my_dict['cwbddata']['resources']['resource']['data']['seaSurfaceObs']  
    records_by_month = a['location']['stationObsTimes']['stationObsTime']
```

轉成 python 程式語言  
原生的資料格式並查表

```
# get header names from any day  
csv_columns = list(flatten_dict(records_by_month[0]).keys())
```

生成 csv 檔案需要的檔頭

```
try:  
    with open(filename_csv, 'wt', newline='') as csvfile:  
        writer = csv.DictWriter(csvfile, fieldnames=csv_columns)  
        writer.writeheader()  
        # records_by_month[0] = the latest (newest) record  
        for record in records_by_month[::-1]:  
            writer.writerow(flatten_dict(record))  
except IOError:  
    print("I/O error :", filename_raw)
```

產生 csv 檔案：

寫檔頭

一行接著一行寫入 (一行 = 一天)

```
print('2. convert xml into csv (complete)')
```

# 3. 壓縮並備份資料

```
# (3) compress and backup 30-day csv files
```

```
now = time.localtime()
date = '{}_{:02d}{:02d}'.format(now.tm_year, now.tm_mon, now.tm_mday)
time = '{:02d}{:02d}{:02d}'.format(now.tm_hour, now.tm_min, now.tm_sec)
FILE_BACKUP = f'cwb_mmc_1month_{date}_{time}.zip'
```

即時生成檔案名稱

```
compression = zipfile.ZIP_DEFLATED
compresslevel = 9
```

設定最高壓縮率

```
with zipfile.ZipFile(FOLDER_BACKUP + FILE_BACKUP, mode='w',
                    compression=compression, compresslevel=compresslevel) as myzip:
```

```
FILES_PATH = FOLDER_CSV + '*.csv'
filenames_current = glob.glob(FILES_PATH)
```

產生 zip 壓縮檔：  
收集要被寫入的檔案 (\*.csv)  
寫入不包含路徑的檔案

```
for filename_current in filenames_current:
```

```
    filename_without_path = os.path.basename(filename_current)
    myzip.write(filename_current, arcname=filename_without_path)
```

```
print('3. back up csv (complete)')
```

# 4. 合併新舊資料

```
# (4) merge csv files and discard additional duplicates
```

```
files_path = FOLDER_CSV + '*.csv'  
filenames_new = glob.glob(files_path)
```

收集最新的 csv 檔案

```
for filename_new in filenames_new:
```

```
    filename_no_path = os.path.basename(filename_new)  
    filename_merge = f'{FOLDER_MERGE}{filename_no_path}'
```

準備舊的 csv 檔案

```
    df_new = pd.read_csv(filename_new)  
    #df_new = pd.read_csv(filename_new, na_values='None')
```

```
    try:
```

```
        df_old = pd.read_csv(filename_merge)  
        #df_old = pd.read_csv(filename_merge, na_values='None')
```

將新和舊的 csv 檔案  
轉成 pandas 的資料結構

```
    except:
```

```
        df_old = None
```

```
    full_df = pd.concat([df_old, df_new])  
    unique_df = full_df.drop_duplicates(subset='dateTime')  
    #unique_df.to_csv(filename_merge) # generate additional column of indices  
    unique_df.to_csv(filename_merge, index=False)
```

根據 dateTime 欄位  
來合併新和舊的紀錄  
並生成合併好的 csv

```
print('4. merge csv (complete)')
```