

民生公共物聯網

資料目錄集

<https://ci.taiwan.gov.tw/dsp/Views/dataset/air.aspx>

使用地震測站

- <https://ci.taiwan.gov.tw/dsp/Views/dataset/earthquake.aspx>

交通部中央氣象署地震監測站 SensorThings API 服務網址

Json

↗ 主網址

↗ 測站資料

↗ 所有地震事件資料，點選各地震事件之 Datastreams 連結，可瀏覽該
地震事件下之所有測站感測資料

API網址

- [https://sta.ci.taiwan.gov.tw/STA_Earthquake_v2/v1.0/Things?\\$expand=Locations&\\$filter=properties/authority%20eq%20%27%E4%B8%AD%E5%A4%AE%E6%B0%A3%E8%B1%A1%E7%BD%B2%27%20and%20substringof\(%27%E5%9C%B0%E9%9C%87%E7%9B%A3%E6%B8%AC%E7%AB%99%27,name\)&\\$count=true](https://sta.ci.taiwan.gov.tw/STA_Earthquake_v2/v1.0/Things?$expand=Locations&$filter=properties/authority%20eq%20%27%E4%B8%AD%E5%A4%AE%E6%B0%A3%E8%B1%A1%E7%BD%B2%27%20and%20substringof(%27%E5%9C%B0%E9%9C%87%E7%9B%A3%E6%B8%AC%E7%AB%99%27,name)&$count=true)

```

{
  "@iot.count": 111,
  "@iot.nextLink": "https://sta.ci.taiwan.gov.tw/STA_Earthquake_v2/v1.0/Things?
$skip=100&$filter=%28%28properties%2Fauthority+eq+%27E4%B8%AD%E5%A4%AE%E6%B0%A3%E8%B1%A1%E7%BD%B2%27%29+and+substringof%28%27%E5%9C%B0%E9%9C%
=true",
  "value": [
    {
      "description": "地震監測站-Donghe-EDH",
      "@iot.id": 2,
      "name": "地震監測站-Donghe-EDH",
      "properties": {
        "authority": "中央氣象署",
        "stationID": "EDH",
        "deviceType": "FBA",
        "stationName": "Donghe"
      },
      "@iot.selfLink": "https://sta.ci.taiwan.gov.tw/STA_Earthquake_v2/v1.0/Things(2)",
      "Locations@iot.count": 1,
      "Locations": [
        {
          "description": "地震監測站-Donghe-EDH",
          "encodingType": "application/vnd.geo+json",
          "@iot.id": 2,
          "location": {
            "type": "Point",
            "coordinates": [121.304, 22.973]
          },
          "name": "地震監測站-Donghe-EDH",
          "@iot.selfLink": "https://sta.ci.taiwan.gov.tw/STA_Earthquake_v2/v1.0/Locations(2)"
        }
      ],
      "TaskingCapabilities@iot.navigationLink": "https://sta.ci.taiwan.gov.tw/STA_Earthquake_v2/v1.0/Things(2)/TaskingCapabilities",
      "Datastreams@iot.navigationLink": "https://sta.ci.taiwan.gov.tw/STA_Earthquake_v2/v1.0/Things(2)/Datastreams",
      "MultiDatastreams@iot.navigationLink": "https://sta.ci.taiwan.gov.tw/STA_Earthquake_v2/v1.0/Things(2)/MultiDatastreams",
      "Locations@iot.navigationLink": "https://sta.ci.taiwan.gov.tw/STA_Earthquake_v2/v1.0/Things(2)/Locations",
      "HistoricalLocations@iot.navigationLink": "https://sta.ci.taiwan.gov.tw/STA_Earthquake_v2/v1.0/Things(2)/HistoricalLocations"
    },
    {
      "description": "地震監測站-Jiqi-EGC",
      "@iot.id": 3,
      "name": "地震監測站-Jiqi-EGC",
      "properties": {
        "authority": "中央氣象署",

```

開啟Colab，利用生成式AI產生Python語法

- 依據以下網址，解析並顯示資料，

[https://sta.ci.taiwan.gov.tw/STA_Earthquake_v2/v1.0/Things?\\$expand=Locations&\\$filter=properties/authority%20eq%20%27%E4%B8%AD%E5%A4%AE%E6%B0%A3%E8%B1%A1%E7%BD%B2%27%20and%20subs](https://sta.ci.taiwan.gov.tw/STA_Earthquake_v2/v1.0/Things?$expand=Locations&$filter=properties/authority%20eq%20%27%E4%B8%AD%E5%A4%AE%E6%B0%A3%E8%B1%A1%E7%BD%B2%27%20and%20subs)



或用ChatGPT產生Python語法

HTTP請求來獲取資料。然後，我們可以解析該JSON格式的資料並將其顯示出來。

以下是一個示例程式碼，演示如何從提供的API端點中獲取資料並顯示它：

```
python Copy code

import requests

# 發送請求並獲取資料
url = "https://sta.ci.taiwan.gov.tw/STA_Earthquake_v2/v1.0/Things?$expand=Locations&$"
response = requests.get(url)

# 檢查請求是否成功
if response.status_code == 200:
    data = response.json()

    # 列印獲取的資料
    print(data)
else:
    print("Failed to retrieve data from the API")
```



這段程式碼會向提供的API端點發送請求，並將獲取到的JSON格式資料列印出來。你可以進一步解析該

```

▶ # prompt: 依據以下網址，解析並顯示資料，https://sta.ci.taiwan.gov.tw/STA\_Earthquake\_v2/v1.0/Things?\$expand=Locations

import requests

url = "https://sta.ci.taiwan.gov.tw/STA\_Earthquake\_v2/v1.0/Things?\$expand=Locations&\$filter=properties/authority%20e"

response = requests.get(url)

if response.status_code == 200:
    data = response.json()
    for item in data["value"]:
        print(item["properties"]["time"])
        print(item["properties"]["locationName"])
        print(item["properties"]["magnitude"])
        print("-----")
else:
    print("Error:", response.status_code)

```

```

-----
SSLCertVerificationError                                Traceback (most recent call last)
/usr/local/lib/python3.10/dist-packages/urllib3/connectionpool.py in _make_request(self, conn, method, url,
body, headers, retries, timeout, chunked, response_conn, preload_content, decode_content,
enforce_content_length)
    467         try:
--> 468             self._validate_conn(conn)
    469         except (SocketTimeout, BaseSSLError) as e:

```

```
(caused by SSLError(SSLCertVerificationError(1, '[SSL: CERTIFICATE_VERIFY_FAILED]
unable to get local issuer certificate (_ssl.c:1007)'))))

During handling of the above exception, another exception occurred:

SSLError                                Traceback (most recent call last)
/usr/local/lib/python3.10/dist-packages/requests/adapters.py in send(self, request,
cert, proxies)
    515         if isinstance(e.reason, _SSLError):
    516             # This branch is for urllib3 v1.22 and later.
--> 517             raise SSLError(e, request=request)
    518
    519         raise ConnectionError(e, request=request)

SSLError: HTTPSConnectionPool(host='sta.ci.taiwan.gov.tw', port=443): Max retries
exceeded with no success
/STA_Earthquake_v2/v1.0/Things?
$expand=Locations&$filter=properties/authority%20eq%20%27%E4%B8%AD%E5%A4%AE%E
(Caused by SSLError(SSLCertVerificationError(1, '[SSL: CERTIFICATE_VERIFY_FAILED]
unable to get local issuer certificate (_ssl.c:1007)'))))
```

Next steps: [Explain error](#)



利用SSL:Error解決問題



1. Install the `certifi` module:

```
!pip install certifi
```

[Use code with caution](#)

2. Import the `certifi` module:

```
import certifi
```

[Use code with caution](#)

在這裡輸入提示



3. Add the `verify` parameter to the `requests.get()` call:

```
response = requests.get(url, verify=certifi.where())
```

[Use code with caution](#)

4. Run the code again.

The `certifi` module provides a set of trusted root certificates for SSL verification. By adding the `verify` parameter to the `requests.get()` call, you are telling `requests` to use the `certifi` bundle to verify the server's certificate. This should resolve the `SSLCertVerificationError`.

提供你對這個答案的評價  



https://data.gov.tw/datasets/search?p=1&size=10&s=_score_desc&rft=%E5%9C%B0%E9%9C%87







顯著有感地震報告

[JSON](#) [XML](#) [API](#)

提供顯著有感地震報告 *112年9月15日起異動下載網址，請於112年12月31日前改接，逾時舊版連結失效。如需大量下載資料者請至氣象資料開放平臺申請會員htt(...詳內)

主要欄位說明: ResourceDesc、MimeType、ProductURL

資料集屬性: 開放資料

   交通部中央氣象署  2024-03-05 11:37:15 更新  335932  1196378  20 [災害防救](#)

https://data.gov.tw/dataset/6068



資料集 ▾

高應用價值主題專區

資料故事館 ▾

互動專區 ▾

消息專區 ▾

諮詢小組

主要欄位說明

*粗體欄位為資料標準欄位

ResourceDesc、MimeType、ProductURL

資料資源下載網址



JSON

檢視資料

等震圖



XML

檢視資料

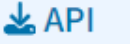
強地動波形



JSON

檢視資料

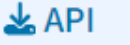
強地動波形



API

檢視資料

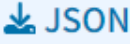
顯著有感地震報告-英文版



API

檢視資料

顯著有感地震報告



JSON

檢視資料

縣市行政區觀測震度資料



XML

檢視資料

等震圖



XML

檢視資料

縣市行政區觀測震度資料

提供機關

交通部中央氣象署

<https://opendata.cwa.gov.tw/api/v1/rest/datastore/E-A0015-001?Authorization=rdec-key-123-45678-011121314>

```
{
  "success": "true",
  "result": {
    "resource_id": "E-A0015-001",
    "fields": ...,
  },
  "records": {
    "datasetDescription": "地震報告",
    "Earthquake": [
      {
        "EarthquakeNo": 113179,
        "ReportType": "地震報告",
        "ReportColor": "綠色",
        "ReportContent": "04/08-13:09臺灣東部海域發生規模5.1有感地震，最大震度花蓮縣和平、宜蘭縣澳花4級。",
        "ReportImageURI": "https://scweb.cwa.gov.tw/webdata/OLDEQ/202404/2024040813093451179_H.png",
        "ReportRemark": "本報告係中央氣象署地震觀測網即時地震資料地震速報之結果。",
        "Web": "https://scweb.cwa.gov.tw/zh-tw/earthquake/details/2024040813093451179",
        "ShakemapImageURI": "https://scweb.cwa.gov.tw/webdata/drawTrace/plotContour/2024/2024179i.png",
        "EarthquakeInfo": ...,
        "Intensity": {
          "ShakingArea": ...
        }
      },
      ...,
      {
        "EarthquakeNo": 113176
```

利用ChatGPT將結構上傳



You

這是API的結構，可否用python讀取資料，並將近一個月內的地震的震央顯示在地圖上

```
{  
  "success": "true",  
  "result": {  
    "resource_id": "E-A0015-001",  
    "fields": [  
      {  
        "id": "ReportType",  
        "type": "String"  
      },  
      {  
        "id": "EarthquakeNo",
```



ChatGPT

以下是使用Python解析並顯示近一個月內地震的震央資料的程式碼：

python

Copy code

```
import requests
import datetime
import folium

# 發送請求並獲取地震資料，忽略SSL憑證的驗證
url = "https://opendata.cwa.gov.tw/api/v1/rest/datastore/E-A0015-001?Authorization=rd
response = requests.get(url, verify=False)

# 檢查請求是否成功
if response.status_code == 200:
    data = response.json()

    # 取得目前時間
    now = datetime.datetime.now()
```

+ 程式碼 + 文字

在下方插入程式碼儲存格

```
import requests
import datetime
import folium

# 發送請求並獲取地震資料，忽略SSL憑證的驗證
url = "https://opendata.cwa.gov.tw/api/v1/rest/datastore/E-A0015-001?Authorization=rdec-key-123-45678-011121314"
response = requests.get(url, verify=False)

# 檢查請求是否成功
if response.status_code == 200:
    data = response.json()

    # 取得目前時間
    now = datetime.datetime.now()

    # 計算一個月前的日期
    one_month_ago = now - datetime.timedelta(days=30)

    # 初始化地震震央的經緯度列表
    earthquake_coordinates = []

    # 解析地震資料，並將近一個月內的地震震央的經緯度加入列表中
    for earthquake in data['records']['Earthquake']:
        origin_time = datetime.datetime.strptime(earthquake['EarthquakeInfo']['OriginTime'], '%Y-%m-%d %H:%M:%S')
        if origin_time > one_month_ago:
            epicenter_latitude = earthquake['EarthquakeInfo']['Epicenter']['EpicenterLatitude']
            epicenter_longitude = earthquake['EarthquakeInfo']['Epicenter']['EpicenterLongitude']
            earthquake_coordinates.append((epicenter_latitude, epicenter_longitude))
```