

API Implementation

API 應用

Python Web crawler



網頁資料擷取

◆ 從網站擷取資料的主要方式有以下幾種：

1. 經由網站，直接下載以某種格式儲存的原始資料檔案，主要以CSV 或JSON格式為主
2. 經由網站提供的專屬API來抓取資料，主要以CSV 或JSON格式為主
3. 經由HTTP爬取網頁資料，並且在本地端進行解析，抽出想要的部份 (網路爬蟲)

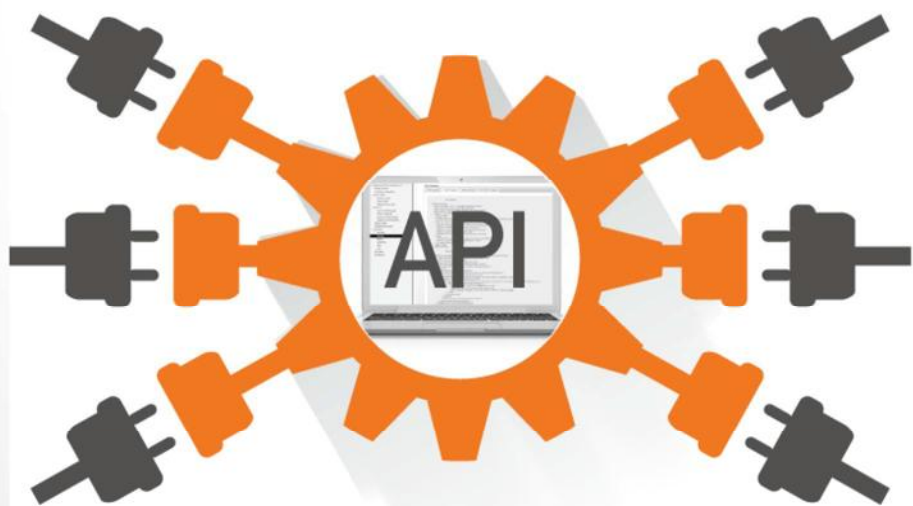
使用API

- ◆ 撰寫爬蟲程式，是為了取得網路上的資料並且進行後續分析，也就是說，獲取資料是我們的目的，寫程式只是手段，因此，如果只是想要完成工作，而不是研究或學習爬蟲程式，在正動手寫程式前，應該先搜尋是否有人已經把你想做的事情做好了。
- ◆ 使用API可以在不用解析網頁架構下就能取得資料。這也是我們首先建議的方式。一來網站的官方API通常附有說明文件，我們能夠藉此了解網站在開放資料上的態度與政策。二來使用API將大幅節省分析網頁及撰寫相關資料提取程式碼的時間，許多時候就算網站沒有提供官方版本，也可以透過開發者工具觀察其內部使用者的API。



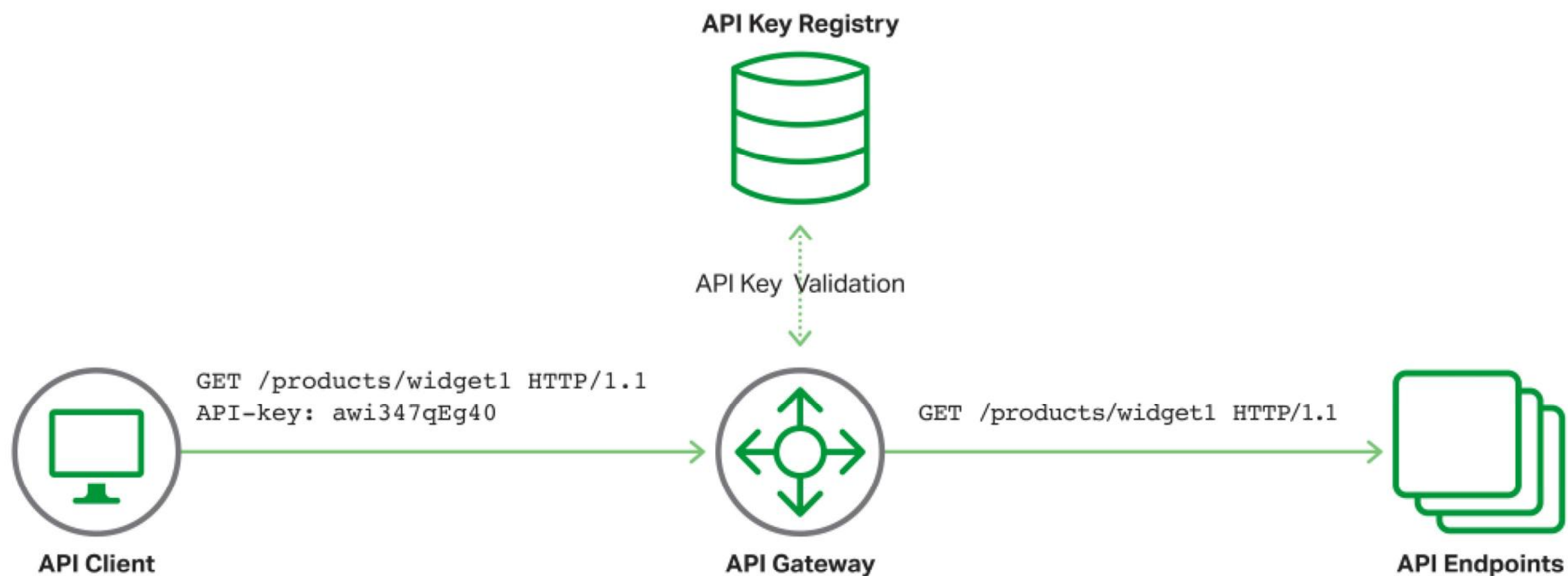
API 簡介

- ◆ 網站官方提供的API(Application Programming Interface)。對一般使用者來說，可以想像為「網站規定好的資料索取方式」，也就是說，這個網站很樂意提供資料，但是它希望透過它規定好的方式來取得。
- ◆ API是一個對於資料供需雙方來說雙贏的方式，一方面身為資料需求方及程式開發者的我們，可以省去處理網頁樣式及排版的力氣，另一方式，對於網站內容提供者最大的好處，就是他們可以透過這個方式去控管每個使用者要資料的流量或頻率。



API 簡介

- ◆ 有些API 需要 API Key 才能使用。API Key/Token 通常是一個亂數字串，你必須先向網站取得此用串作為身份證明，在程式與網站溝通的過程中，都必須附帶API Key以表明你的身份，讓該網站就可以透過此字串辨識你，進一步控管你索取資流量或頻率等。



API 簡介

- ◆ 即使是透過程式與網站溝通，就要遵循網路上的通訊協定與方法，主要常見的HTTP方法有以下二種：
 - **GET**：直接使用資源位址以取得資料。我們可以簡單地把這個方法想成：只要在瀏覽器內直接貼上網址（及參數），按下**Enter**後，網站就會把資料回傳給你。
 - **POST**：在背景送出資料到某個位址，希望該處的**Server** 能夠處理。有時候我們在瀏覽網頁時，直接上網址並沒有辦法取得你想要的資料，你必須跟網站有些互動，例如填寫並送出表單之後，才能夠取得資料，你也可能發現在過程中瀏覽器的網址列沒有變化，這就非常可能是透過**POST**方法。實務上90%的情況我們可以透過**GET**或**POST**方法取得資料。
 - 透過HTTP方法送出資訊給server之後，**Server**回應的資訊有兩種最常見的格式：**JSON**與**XML**

API 簡介

- ◆XML 格式的網址回傳結果如下。這是早期的網路資料交換格式，其語法較為豐富但也比較笨重，目前多被輕量的JSON格式取代，不過有些網站為了向前相容以前的程式，也會提供XML的回傳格式。

<http://www.omdbapi.com/?t=iron+man&apikey=433e8713&r=xml>

```
<root response="True">
  <movie title="Iron Man" year="2008" rated="PG-13" released="02
May 2008" runtime="126 min" genre="Action, Adventure, Sci-Fi"
director="Jon Favreau" writer="Mark Fergus (screenplay), Hawk
Ostby (screenplay), Art Marcum (screenplay), Matt Holloway
(screenplay), Stan Lee (characters), Don Heck (characters),
Larry Lieber (characters), Jack Kirby (characters)"
actors="Robert Downey Jr., Terrence Howard, Jeff Bridges,
Gwyneth Paltrow" plot="After being held captive in an Afghan
cave, billionaire engineer Tony Stark creates a unique
weaponized suit of armor to fight evil." language="English,
Persian, Urdu, Arabic, Hungarian" country="USA"
awards="Nominated for 2 Oscars. Another 20 wins & 65
nominations." poster="https://m.media-
amazon.com/images/M/MV5BMTczNTI2ODUwOF5BMl5BanBnXkFtZTcwMTU0NTIz
metascore="79" imdbRating="7.9" imdbVotes="817,761"
imdbID="tt0371746" type="movie"/>
</root>
```

API 簡介

- ◆ JSON格式的範例如下，我們可以把JSON文件簡單地想成Python的dictionary，它就是一堆鍵與值的對應(key-value pairs)。很明顯地，這種格式處理起來比HTML網頁文件簡單許多。

`https://testblog.com/api/v4/users/9487/posts?api_key=YOUR_API_KEY&from =08012016?to=12312016&format=json`

- ◆ 使用GET方法的API，首先開頭當然是網站的網址，接著可能是一些有意義的路徑目錄，如API版本等，也可能依照網站內容有不同的目錄名稱，如blog網站可能有user、post等目錄，接著多半需要在參數中附上API Key或Token，以及相關的參數(如blog貼文的起訖日期等)。
- ◆ 我們之後將透過多個範例介紹如何使用API。我們將以JSON格式為主，關於XML檔案的讀取會在部份範例上呈現。

API 應用 - 新北市電影名冊(CSV)

```
9 import requests as rq #戴入requests 套件，縮寫rq
10 import csv #戴入csv套件，以處理csv格式
11 import pandas as pd #戴入pandas套件，縮寫為pd
12
13 #開放資料：新北市電影院名冊
14
15 url='http://data.ntpc.gov.tw/od/data/api/61c99F42-8A90-4ADC-9C40-BA9E0EA097AA?format=csv'
16
17 r=rq.request('GET',url) #對url發出get請求
18
19 #將csv格式的字串轉換成二維串列
20 data=list(csv.reader(r.text.split('\n')))
21
22 #將資料集串列轉換成DataFrame
23 df=pd.DataFrame(data[1:len(data)-1], columns=['名稱','地址','電話號碼','廳數'])
24 df.index+=1 #資料順序從1開始
25 print(df)
```

	名稱	地址	電話號碼	廳數
1	幸福影城	三重區三和路4段163巷12號	22865540	6
2	天台影城	三重區重新路2段78號4樓	29787700	5
3	林園電影城	板橋區府中路175號3樓	29605333	3
4	華麗電影院	板橋區府中路175號5樓	29605333	2
5	鴻金寶	新莊區民安路188巷5號4樓	22070222	5
6	中和國賓	中和區中山路3段122號4樓	22268088	7
7	威秀影城	板橋區新站路28號10樓	77386608	9
8	板橋秀泰	板橋區縣民大道二段3號2、3、4樓	29685588	17
9	林口威秀	新北市林口區文化三路一段356號3樓、4樓	87801166	9
10	新莊國賓	新莊區五工路66號3、4F	85216517	13
11	林口國賓	林口區文化三路一段402巷2號4樓	26080011	8

API 應用 - 新北市電影名冊(JSON)

```
1 import requests as rq #載入requests 套件，縮寫rq
2
3 #開放資料：'YouBike 臺北市公共自行車即時資訊'
4 url='http://data.ntpc.gov.tw/od/data/api/54DDDC93-589C-4858-9C95-18B2046CC1FC?$format=json'
5
6 html_content=rq.get(url) #向html提出get請求
7 json_data=html_content.json() #將回傳內容轉換成json格式
8
9 #item_detail是tuple 的第二個元素，型態為字典
10 for item_detail in json_data:
11     print_info = '站點：'+item_detail['sna']+', '+\
12                 '地址：'+item_detail['ar']+', '+\
13                 '總停車格：'+item_detail['tot']+', '+\
14                 '場站目前車輛數量：'+item_detail['sbi']+', '+\
15                 '空位數量：'+item_detail['bemp']+', '+\
16                 '資料更新時間：'+item_detail['mday']
17     print(print_info) #顯示結果
```


API 應用 - 新北市電影名冊(JSON)

站點：海洋公園,地址：大義路/學勤路口(西北側),總停車格：36,場站目前車輛數量：5,空位數量：29,資料更新時間：20181120233935
站點：浮洲合宜住宅(合宜一路),地址：大觀路二段265巷/合宜一路(東南側),總停車格：32,場站目前車輛數量：16,空位數量：16,資料更新時間：20181120233936
站點：文山國中,地址：北宜路一段118號(對面),總停車格：30,場站目前車輛數量：3,空位數量：27,資料更新時間：20181120233931
站點：臺北大學(資訊中心),地址：大學路151號(資訊中心旁),總停車格：56,場站目前車輛數量：10,空位數量：46,資料更新時間：20181120233934
站點：捷運府中站(1號出口),地址：縣民大道一段/府中路口(西南側),總停車格：76,場站目前車輛數量：9,空位數量：65,資料更新時間：20181120233934
站點：新莊田徑場,地址：復興路一段209號(對面),總停車格：42,場站目前車輛數量：22,空位數量：20,資料更新時間：20181120233937
站點：新泰國中,地址：新泰路359號(旁),總停車格：30,場站目前車輛數量：2,空位數量：28,資料更新時間：20181120233946
站點：碧潭渡船頭,地址：新店路36號(對面),總停車格：32,場站目前車輛數量：25,空位數量：7,資料更新時間：20181120233917
站點：山佳火車站,地址：中山路三段108號,總停車格：32,場站目前車輛數量：3,空位數量：29,資料更新時間：20181120233938
站點：新北市勞工活動中心,地址：五工六路9號(前廣場),總停車格：34,場站目前車輛數量：3,空位數量：30,資料更新時間：20181120233939
站點：板橋重慶公園,地址：重慶路276號(對面),總停車格：40,場站目前車輛數量：11,空位數量：28,資料更新時間：20181120233940
站點：林口社區運動公園(公園路),地址：公園路192號(旁),總停車格：40,場站目前車輛數量：19,空位數量：20,資料更新時間：20181120233938
站點：山北公園,地址：莊敬路33巷36弄2號(對面),總停車格：32,場站目前車輛數量：13,空位數量：17,資料更新時間：20181120233915
站點：景平大勇街口,地址：景平路127號(旁),總停車格：38,場站目前車輛數量：21,空位數量：15,資料更新時間：20181120233946
站點：中港大排願景館,地址：中央路301號(對面),總停車格：30,場站目前車輛數量：10,空位數量：18,資料更新時間：20181120233941
站點：頭前國小,地址：頭前路93號(對面),總停車格：30,場站目前車輛數量：5,空位數量：24,資料更新時間：20181120233935
站點：淡金北新站,地址：淡金路一段547號(前),總停車格：46,場站目前車輛數量：7,空位數量：38,資料更新時間：20181120233917
站點：忠孝活動中心,地址：忠孝街26巷8號(前),總停車格：28,場站目前車輛數量：18,空位數量：6,資料更新時間：20181120233935
站點：淡水海關碼頭園區,地址：中正路261號(旁),總停車格：40,場站目前車輛數量：17,空位數量：22,資料更新時間：20181120233931
站點：縣民漢生東路口,地址：縣民大道二段112號(前),總停車格：60,場站目前車輛數量：6,空位數量：53,資料更新時間：20181120233941
站點：時光公園,地址：華江陸橋旁/文化路二段北側公園口,總停車格：40,場站目前車輛數量：7,空位數量：33,資料更新時間：20181120233939
站點：積穗國中,地址：民安街71號(對面),總停車格：32,場站目前車輛數量：19,空位數量：13,資料更新時間：20181120233934
站點：員和公園,地址：中央路二段66號(對面),總停車格：34,場站目前車輛數量：24,空位數量：10,資料更新時間：20181120233943
站點：環河公園,地址：南雅西路二段301巷41弄5-5號(旁),總停車格：36,場站目前車輛數量：14,空位數量：22,資料更新時間：20181120233927
站點：中興中正路口,地址：中興路二段41號(前),總停車格：30,場站目前車輛數量：12,空位數量：17,資料更新時間：20181120233926
站點：新莊國民運動中心,地址：公園路66號(對面),總停車格：40,場站目前車輛數量：31,空位數量：9,資料更新時間：20181120233918
站點：佳和公園(中山路二段64巷),地址：中山路二段64巷7弄15-4號(對面),總停車格：36,場站目前車輛數量：20,空位數量：16,資料更新時間：20181120233921
站點：三鶯國民運動中心,地址：文化路164號(旁),總停車格：32,場站目前車輛數量：9,空位數量：22,資料更新時間：20181120233932
站點：崇林國中,地址：麗園路121號(對面),總停車格：34,場站目前車輛數量：1,空位數量：32,資料更新時間：20181120233924
站點：成泰路一段(中油),地址：成泰路一段194號(對面),總停車格：26,場站目前車輛數量：21,空位數量：5,資料更新時間：20181120233930

API Key應用 - IMDB API

- ◆ IMDB的非官方API是一個比較複雜且完整的API。
- ◆ 之所以說是非官方，是因為IMDB本身並沒有提供官方的API，只有定期提供資料庫檔案的打包下載。
- ◆ 不過有一網站會下載官方資料庫，並製作非官方的API供外界使用。本例子OMDB網站便是一個例子。
- ◆ 我們將說明如何透過OMDB API搜尋所有iron man(鋼鐵人)相關影片，並計算統計資料(例如發行年份分與平均評分)。

API Key應用 - IMDB API

- ◆ OMDB 網站是網路上較知名的IMDB非官方API，其網址為 <http://www.omdbapi.com>。
- ◆ 網頁首頁列出了API使用方式及各種查詢參數。在下方提供網頁界面，讓使用者直接操作、測試參數的意義。例如在介面上搜尋電影標題「iron man」，會告訴你API的網址內容，以及回傳的結果。
- ◆ 我們可以看到回傳的內容就是一個JSON檔案，包含許多該電影的資訊，如年份，演員、評分等。
- ◆ 此介面也可供以IMDB ID查找電影的方式，例如在介面輸入2008年鋼鐵人電影的ID「tt0371746」，介面一樣會回傳相同的結果。
- ◆ 中文電影也是可以搜尋的，只要知道電影的英文片名或IMDB ID即可。

OMDb API
The Open Movie Database

API Key應用 - IMDB API

- ◆ 測試區提供的介面有幾點需要注意，首先其顯示的API網址內容並未包含API Key欄位，真正使用API時必須一併附帶API Key。
- ◆ 其次，「以標題查找」的方式只會回傳第一筆找到的資料，只此我們會說明如何使用其它參數搜尋。
- ◆ 最後，我們可以試著搜尋不存在的片名，看它的回傳值，我們會發現若結果不存在時，其回傳的JSON檔案中的Response會是False，因此我們可以檢查Response的值來確認搜尋結果是否存在。

API Key應用 - IMDB API

◆ 要搜尋OMDB API，必須先取得API Key。OMDB的API Key有兩種方式可以取得：

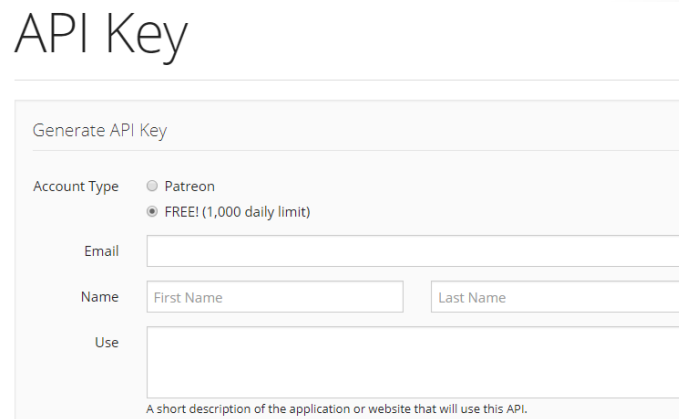
1. 付費取得不受限制的API Key。
2. 免費索取API Key，但每日有1000次的存取限制。

◆ 我們選擇免費的API Key。首先在OMDB網頁點選上方「API Key」選項，或到以下網址：

<http://www.omdbapi.com/apikey.aspx>

◆ 在該網頁點選「FREE！」，並輸入Email、姓名、用途等(隨意輸入即可)，就能夠在Email信箱收到免費的API Key，作為後續呼叫API時使用。

API Key



The screenshot shows a web form titled "Generate API Key". It has two radio buttons for "Account Type": "Patreon" and "FREE! (1,000 daily limit)". The "FREE!" option is selected. Below this are three input fields: "Email", "Name" (split into "First Name" and "Last Name"), and "Use" (a larger text area). At the bottom, there is a small text label: "A short description of the application or website that will use this API."

API Key應用 - IMDB API

- ◆ 有了API Key，就能開始使用API。測試區沒有提供使用關鍵字搜尋多部電影的功能，但我們可以從網頁的參數說明得知要使用參數s，並使用+號來串接關鍵字，例如要搜尋鋼鐵人的所有電影或影集，可使用API 如下：

[http://www.omdbapi.com/?s=iron+man&apikey=\[YOUR_API_KEY\]](http://www.omdbapi.com/?s=iron+man&apikey=[YOUR_API_KEY])

API Key應用 - IMDB API

- ◆ 搜尋結果如右圖，回傳的結果主要分成兩部份，第一部份是鍵值為 **Searchd**的區塊，其值為前10筆搜尋結果的list，第二部份鍵值為**TotalReults**，其值為搜尋結果筆數。我們可以看到此處有82筆搜尋結果，而第一次回傳前10筆。
- ◆ 若要取得第11筆之後的資料，只要加上參數 **page=[PAGE_NUMBER]**即可，例如**page=3** 就是第21到30筆的資料，而第9頁就是第81及82筆資料。
- ◆ 每一筆搜尋結果的欄位有：**Title**(片名)、**Year**(發行份)、**imdbID**等，在此我們感興趣的只有電影的**imdbID**，因為我們需要用 **imdbID**去進一步搜每部影片，以取得各影片的詳細資料如評分等。

```
{
  "Searchd": [
    {
      "Title": "Iron Man vs Batman",
      "Year": "2010",
      "imdbID": "tt3426078",
      "Type": "movie",
      "Poster": "N/A"
    },
    {
      "Title": "Iron Man 3: Advancing the Tech",
      "Year": "2013",
      "imdbID": "tt3455774",
      "Type": "movie",
      "Poster": "N/A"
    },
    {
      "Title": "Sharon Israel's Iron Man",
      "Year": "2014",
      "imdbID": "tt3464750",
      "Type": "movie",
      "Poster": "N/A"
    }
  ],
  "totalResults": "82",
  "Response": "True"
}
```


API Key應用 - IMDB API

程式進行的流程為：

1. 用關鍵字搜尋所有相關影片，記下每部影片的imdbID：

首先搜尋關鍵字，將imdbID放入m_ids中

2. 用imdbID進一步搜尋所有影片，紀錄每部影片的詳細資訊：

搜尋m_ids中的每一個imdbID，將電影資料放入movies內

3. 顯示統計結果(發行年份分佈與平均評價)：

針對每一部電影，取得其發行年份(Year欄位)，及評分(imdbRating欄位)。此處計算年份分佈的方式是利用

Python內建的統計函式, `collections.Counter()`。其輸入是一個list，輸出是list裡面各個項目的出現次數，並由大到小排序。

API Key應用 - IMDB API

1. 首先定義`get_data()`以取得OMDB API的回傳資料。因為API的回傳值是JSON檔，我們直接讀入、檢查其Response欄位是否為True，若為True表示結果正常，若為False則回傳False，代表沒有資料。
2. 接著定義`search_ids_by_keyword()`。
3. 所有取得的電影imdbID都會放在`movie_ids list`中回傳。方式是先取得第一頁的10筆資料，並計算總頁數(例如82筆資料就是9頁)，接著在網址附上page資訊依序取得11筆以上的資料。
4. 接著定義`search_by_id()`取得每一筆電影的詳細資訊。`search_by_id()`的回傳結果是JSON object，也就是dict。

Q & A