**透過API爬行**

JavaScript一向是爬行程序的阻礙。過去有過一段時間，對網頁伺服器的請求一定會與瀏覽器看到的一樣。

隨著JavaScript與Ajax普及，這種狀況變得罕見。第11章討論使用Selenium取得資料。這是簡單的做法。他幾乎可行。

這一章完全略過JavaScript並直接從資料元抓:產生資料的API

**API簡介: API定義軟體間的標準語法**

**針對網頁伺服器與瀏覽器溝通的API，**但一班API不限於 ”跨網際網路” 或網頁技術而已，網頁API通常用於眾所周知且有文件的公開服務。舉例來說，ESPN有個體育資訊的API。Google([https://console.developers.google.com](https://console.developers.google.com/projectselector/apis/dashboard?pli=1&supportedpurview=project&project=&folder=&organizationId=))有很多給開發者的API。

API通常以JSON與XML格式回傳。

API回傳JSON例子:

{“user”:{“id”:123 , “name”: “Ryan Mitchell”, “city”: “Boston”}}

API回傳XML例子

<user><id>123</id> <name>Ryan Mitchell</name><city>Boston</city></user>

FreeGeoIP(http://greegeoip.net)有個簡單的API可以轉換IP為指成十件地址，你可以在瀏覽器上測試API的請求:

<http://greegeoip.net/json/50.78.253.58>

可改變參數請求XML或CSV

<http://greegeoip.net/xml/50.78.253.58>

<http://greegeoip.net/csv/50.78.253.58>

**HTTP方法與API**

前面看過API以GET像伺服器請求資訊。透過HTTP 像網頁伺服器請求資訊的方式(或方法)主要有四種

* GET
* POST
* PUT
* DELETE

技術上不只4種(例如: HEAD、 OPTIONS、CONNECT)，但較少用於API。大部分API都只使用4種或是更少。

**GET**

以瀏覽器的位址列 造訪網站實用的方法。可以當作是:”嘿!伺服器，請給我這個資訊”

**POST**

提交資訊給伺服器。登入網站時發出的POST請求加上使用著名稱與(可能)加密過的密碼。以API發出POST請求的意思是:”請儲存資訊到資料庫”

**PUT**

常用於互動網站，偶爾用於API。PUT請求用於更新物件或資訊。舉例來說，一個API可能POST建構新使用者，但需要以PUT更新使用者的郵件地址。

**DELETE**

用於刪除物件。舉例來說，若向<http://myapi.com/user/23>發出DELETE的請求，他可能會刪除ID為23的使用者。DELETE很少用於API，他主要用於傳播資訊或張貼資訊而非讓使用者刪除資訊。

**API的回應:**

如前面的FreeGeoIP，API的功能之一是良好格式的回應。最常見的回應格式XML(eXtensible Markup Language)、JSON(JavaScript Object Notation)

JSON比XML更受歡迎，過去PHP或.NET等伺服器端腳本常作為API的端點。現在較多是Angular或Backbone等框架收發API呼叫。伺服器端技術通常與資料格式無關，但Backbone等JavaScript函示庫處裡JSON更方便。

API的回應類型是開發者的想像力而定。CSV也是典型的回應(例如FreeGeoIP)。

有些API甚至會產生檔案。有些伺服器會產生加文字的圖檔、XLSL、或PDF檔案。

有些API完全不回應。舉例來說:若像伺服器請求建立新的發文，他可能只以HTTP回應馬200表示 “發文了 ； 沒問題 ! ”。有些可能只是這樣回應

{“success” : “true”}，若發生錯誤，你可能收到這樣的回應:

{“error” : ” Something super bad happened”}

向API請求時，做好先檢查是否為JSON(或XML、CSV等預期的格式)

**解析JSON**

各種類型的API與運作方式，定檢視這些API的JSON回應。

JSON解析函式庫是Python的核心函式庫之一。**只需要在上面輸入import json就好!!!**不像其他語言可能將JSON解析成JSON物件或JSON節點，Python更彈性的**將JSON物件轉換成字典、JSON陣列轉換為清單、JSON字串轉換為字串等**

這種方式非常容易存取與操作JSON中的值。

**沒有文件的API**

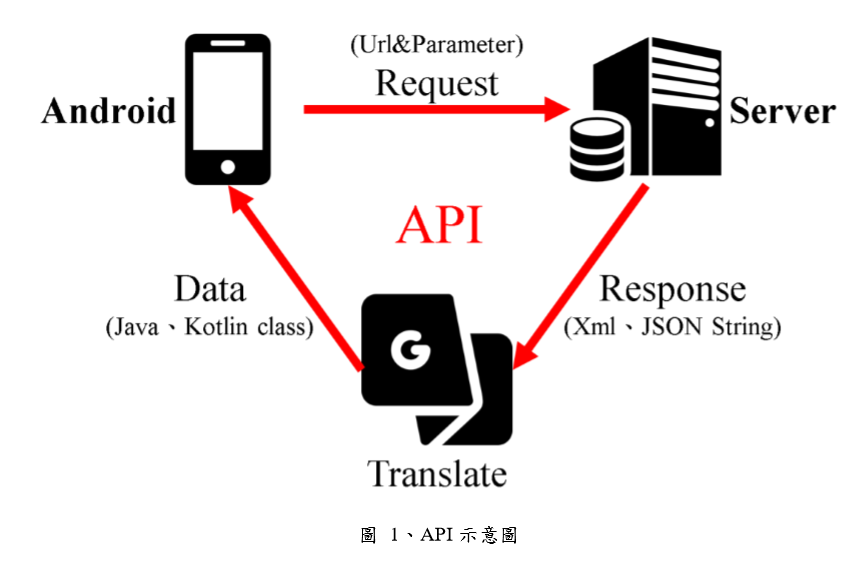
有文件的API，他的開發者是希望讓公眾使用，公開他的資訊並假設其他開發者會使用。但大部分的API都並未完全公開。

為甚麼會建構不公開的API? 如前述，跟JavaScript有關。

傳統上，動態網站的網頁伺服器在使用者請求網頁時有幾個任務。

1. 處裡GET請求
2. 依資料庫讀取資料
3. 將資料轉換成HTML
4. 發送HTML給使用者

HTML方法



Python

隨著JavaScript框架普及，許多產生HTML的任務由伺服器轉移到瀏覽器上。伺服器發送HTML模板給使用者的瀏覽器，由另一個Ajax情求仔入內容並放到HTML模板中。這些都發生在瀏覽器/用戶端。

這對網站擷取程序產生了問題，以前發出**HTML網頁請求會得到HTML網頁與完整的內容。現在只會得到沒有內容的HTML模板。**

**解決**: Selenium用於解決這個問題。現在程式設計師的網站擷取程序變成瀏覽器，請求html模板、執行JavaScript、載入資料、然後才擷取網頁資料。由於HTML都完成載入，他基本上簡化成前面解決過的問題---解析現有的HTML(第2、3章)

Selenium會把使用者不關心的”多於資料”也載入。呼叫追蹤程式、載入廣告、呼叫廣告的追蹤程式。圖像、CSS、第三方自行資料---全部都會載入。

解決:由於伺服器不再將資料轉換成HTML，他們通常只是資料庫的簡單包裝。這些包裝紙是將資料庫中的資料透過API傳送到網頁。

當然這些API的本意用於自己的網頁，因此開發者不會寫文件並假設沒有人會注意到。(但它就在那)

實作:

舉例來說: New York Times的網站(<http://nytimes.com>)透過JSON載入搜尋結果。造就此連結:

<http://query.nytimes.com/search/sitesearch/#/python>

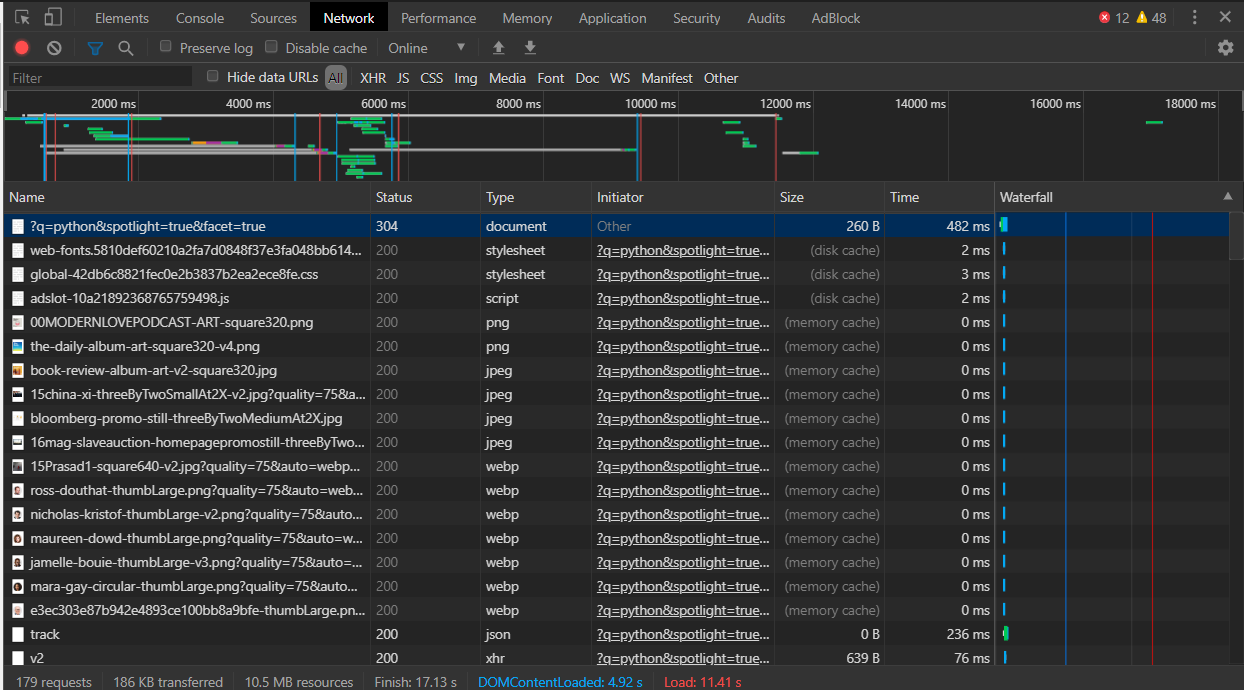
它會顯示”python的搜尋結果”。使用urllib或Request函式庫解析此頁不會看到任何搜尋結果。他透過另一個API呼叫載入:

<http://query.nytimes.com/scv/add/v1/sitesearch/.json?q=python&spotlight=true&facet=true>

若想要用Selenium載入此頁，每個搜尋必須發出100個請求與下載600-700kB資料。直接使用API僅發出一個請求與60kbb格式簡單的資料

**找出無文件API**

開啟Chrome的Inspector視窗並點擊Network分頁，如圖



請注意，你必須在網頁載入前開啟此視窗，才會看到過程。

打開後你會即時看到瀏覽器為顯示網頁所發出的呼叫。它包括API呼叫。

找出無文件API就像是調查工作(“有自動化尋找API的技術”)

網路呼叫中的 **API呼叫** 有些方便搜尋的特徵

* 通常具有JSON或XML。你可以用search/filter過濾清單
* GET請求的URL會帶有參數值。舉例來說，若要尋找 搜尋結果或**載入資料的API呼叫**，之需要過濾你的搜尋關鍵字、網頁ID、或其他資訊
* 類型通常是XHR

要找尋的 **發出呼叫的API** 不一定很明顯(熟能生巧吧)

**無文件API的文件**

找出**發出呼叫的API**，寫一些文件通常有用，特別是擷取程序依賴此呼叫時。你可能會想要載入該網站的多個網頁，從網頁分頁過濾出目標API呼叫，這麼做可以看到各個網頁的呼叫並識別它 接受回傳的欄位。

每個API呼叫可依下列欄位識別寫文件:

* HTTP方法
* 輸入

--路徑參數

--標頭(包括Cookie)

--內容(PUT與POST呼叫)

* 輸出

--回應標頭(包括Cookie)

--回應內容型別

--回應內容欄位

**自動化尋找API與寫文件**

寫API確實很麻煩。有些網站刻意隱藏運作方式史的他更麻煩，而寫API大部分都是重複的任務。

在<http://github.com/REMitchell/apiscraper> 有程式客可以解決基本任務。