



國家發展委員會

109年政府資料開放平臺
資料品質訓練手冊

API標準化與OAS規範介紹

大綱

壹

API介紹及應用

貳

API規範文件

參

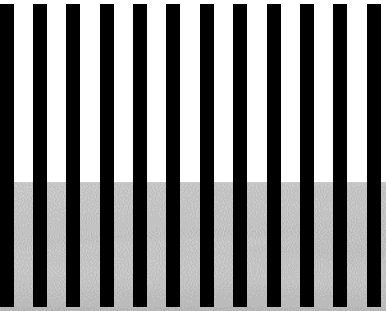
OAS介紹與驗證方式

肆

常見問答

API介紹及應用

#01



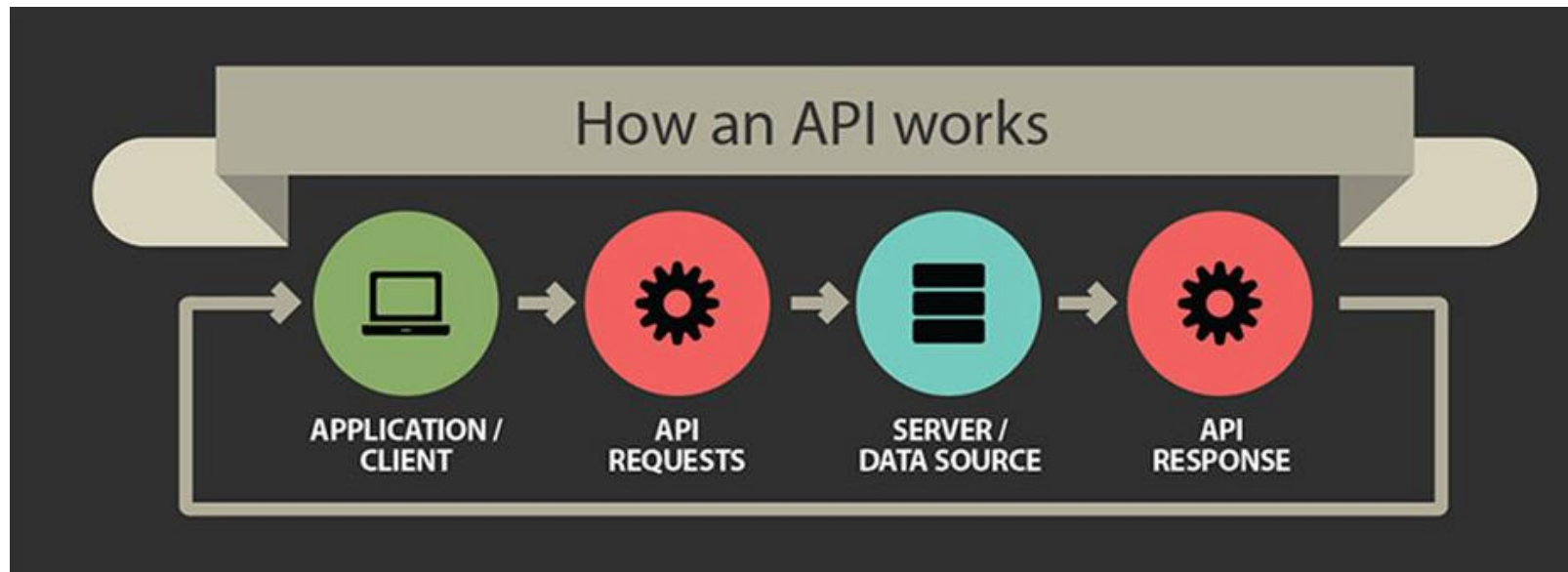


API介紹及應用

1) 應用程式介面(API) 介紹

1-1) 什麼是API?

- 應用程式介面 (Application Programming Interface, API)。
- API 是程式/系統/軟體 彼此相互銜接、溝通的約定方式(橋樑)。
- 透過 API 的通道，不同程式/系統/軟體得以相互取用部分功能或元件，也彼此交換資料內容。



1-1) 什麼是API?

誰是 API 提供者

因應網路平臺的多樣性的發展，許多大型網站逐漸開放物件功能和資訊與其它網站進行串接、共享

而大型網站將這些開放分享的資訊功能，開發打包成可物件連結的應用程式介面，提供其它開發者進行應用串接





1-1) 什麼是API?

API提供者之一：政府相關單位

- 推動Open Data：開放政府資料供給一般人做使用分析
- 如何更便利的讓使用者取得資料→ Open API
- 提供第三方使用者可透過API取得各式結構化資料



1-1) 什麼是API?

API提供者之一：政府相關單位

- 規劃 REST Web API，讓資料使用者可以 HTTP GET 方式，取得政府資料開放平臺之資料

政府資料開放平臺API擴充專區

為便利民眾取用政府開放資料，擴充政府資料開放平臺API功能，

提供第3方使用者可透過API取得本平臺動態產生之各式結構化資料，本平臺API擴充項目如下：

(一)各地方政府資料開放情形：https://data.gov.tw/v1/export/local_open

(二)各部會資料開放情形：https://data.gov.tw/v1/export/central_open

(三)全部資料集清單：<https://data.gov.tw/datasets/export/csv?type=dataset&order=pubdate&q=&uid=>

(四)已下架資料集清單：https://data.gov.tw/v1/export/unpub_dataset

(五)資料分類統計

- 資料集統計：<https://data.gov.tw/statistics/datasets/datasets/export/json?type=datasets>

- 主題分類統計：<https://data.gov.tw/statistics/datasets/topic/export/json?type=topic>

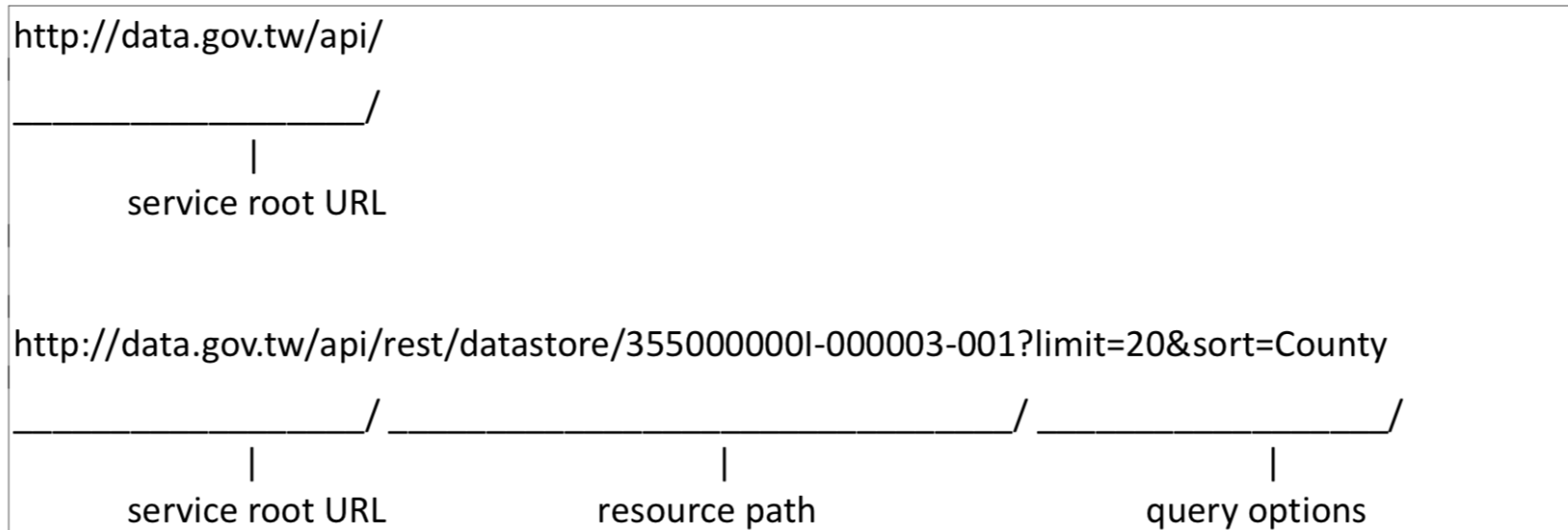


1-2) API儲存/更新/刪除/取得資料機制(1)

- API有多種製作方式，其中包含REST。
- REST為軟體架構設計風格，若使用REST的方式傳輸API，則稱為RESTful API。
- RESTful API由三種元件組成：
 - Nouns名詞：定義資源位置的 **URI**，每個資源在網路上都會有唯一的位置，就如每戶人家都有唯一的地址一樣。
 - Verbs動詞：對**伺服器端要做的動作**。
 - Content Types 資源呈現方式：API 傳遞內容可以以多種方式表現，最常用的是 **JSON檔案格式**。

1-2) API儲存/更新/刪除/取得資料機制(2)

- 定義資源位置的 **URI**命名原則：
 - 一般資源用複數名詞，如 /cars 或 /cars/123，而不用 /car/ 或 /car/123
 - 唯一資源（亦即對client而言只有一份的資源）用單數名詞，如：/user/info，其中user是指目前登入的使用者，所以用單數
 - 資源的層級架構，可反應在API endpoint設計上，如：/books/123/ch/2
 - Utility API 與 resource API 性質不同，它的 endpoint 設計只要合理即可，例如 /search?q={keywords}





1-2) API儲存/更新/刪除/取得資料機制(3)

- 對伺服器端要做的動作：
 - GET: 讀取資源
 - PUT: 替換資源
 - DELETE: 刪除資源
 - POST: 新增資源
- JSON檔案格式了解：
 - 大部份API使用JSON檔案格式進行傳輸及溝通。
 - JavaScript物件表示法(JSON)為一個資料交換格式，各種不同程式語言皆可讀取/寫入。

1-2) API儲存/更新/刪除/取得資料機制(4)

- JSON用於描述資料結構，有兩種結構：

- 物件 (object)

- 以 `{ }` 作為開始與結束，內容形式為 key & value，並用 `:` 分隔
`{key : value}`

- 陣列 (array)

- 以 `[]` 作為開始與結束，內容形式為一組值的集合，值的形式有：

- 字串 (string，以 `" "` 包住)
- 數值 (number)
- 物件 (object)

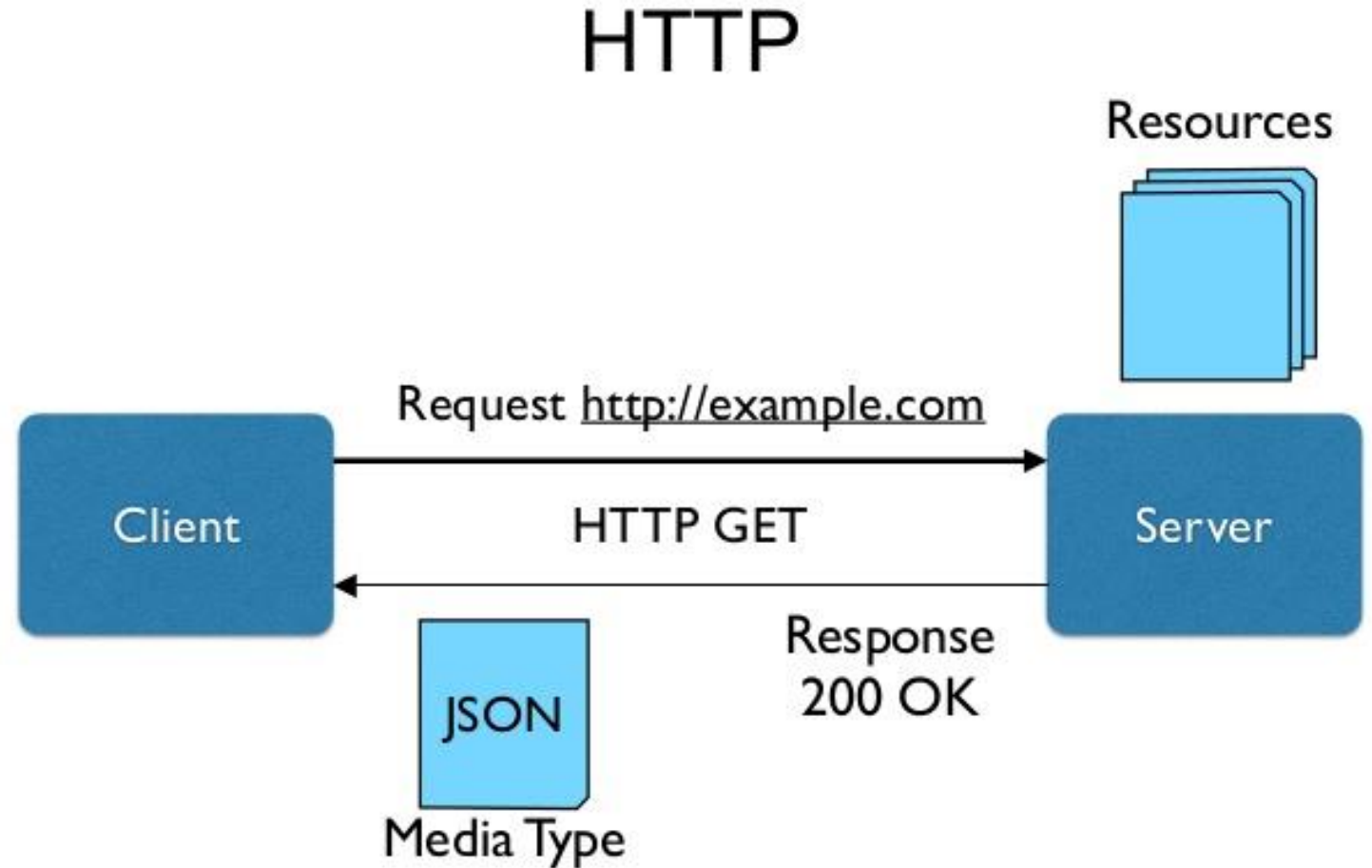
`["string", 123, {key : value}, [1,2,3,4,5]]`

資料集公共政策網路參與平台-提點子 JSON檔案

```
1 [
2   {
3     "publishDate" : "2015-09-10 13:26:19",
4     "網址" : "http://join.gov.tw/idea/",
5     "標題" : "Join 平台應提供匯出資料供批次下載",
6     "提議內容" : "目前 Join 平台為方便機關人員作業。",
7     "利益與影響" : "這是 kiang 在 gov 提出的想法。",
8     "附議數量" : "22",
9     "附議門檻" : "250",
10    "提送日期" : "2015-09-10 17:12:05",
11    "關注數量" : "1",
12    "留言數量" : "0",
13    "googleAnalytics" : "無",
14    "提議者" : "au"
15  },
16  {
17    "publishDate" : "2015-09-10 16:08:10",
18    "網址" : "http://join.gov.tw/idea/",
19    "標題" : "你是否贊成推動「十八歲投票權」及「二十歲被選舉權」?",
20    "提議內容" : "鑒於國民年滿十六歲即可工作、納稅，
21    "利益與影響" : "世代正義是我國民主發展所必須正視的課題，
22    "附議數量" : "0",
23    "附議門檻" : "0",
24    "提送日期" : "2015-09-10 16:11:31",
25    "關注數量" : "0",
26    "留言數量" : "0",
27    "googleAnalytics" : "無",
28    "提議者" : "森里蚩一"
29  }
30 ]
```

1-2) API儲存/更新/刪除/取得資料機制(5)

1. API的URI位置：索取資源的地方(跟“誰”要東西，或給“誰”東西)。
2. API的動作：要做的事情(跟誰“要”東西，或“給”誰東西)。
3. API的JSON檔案：做事情的內容(跟誰要“東西”，或給誰“東西”)



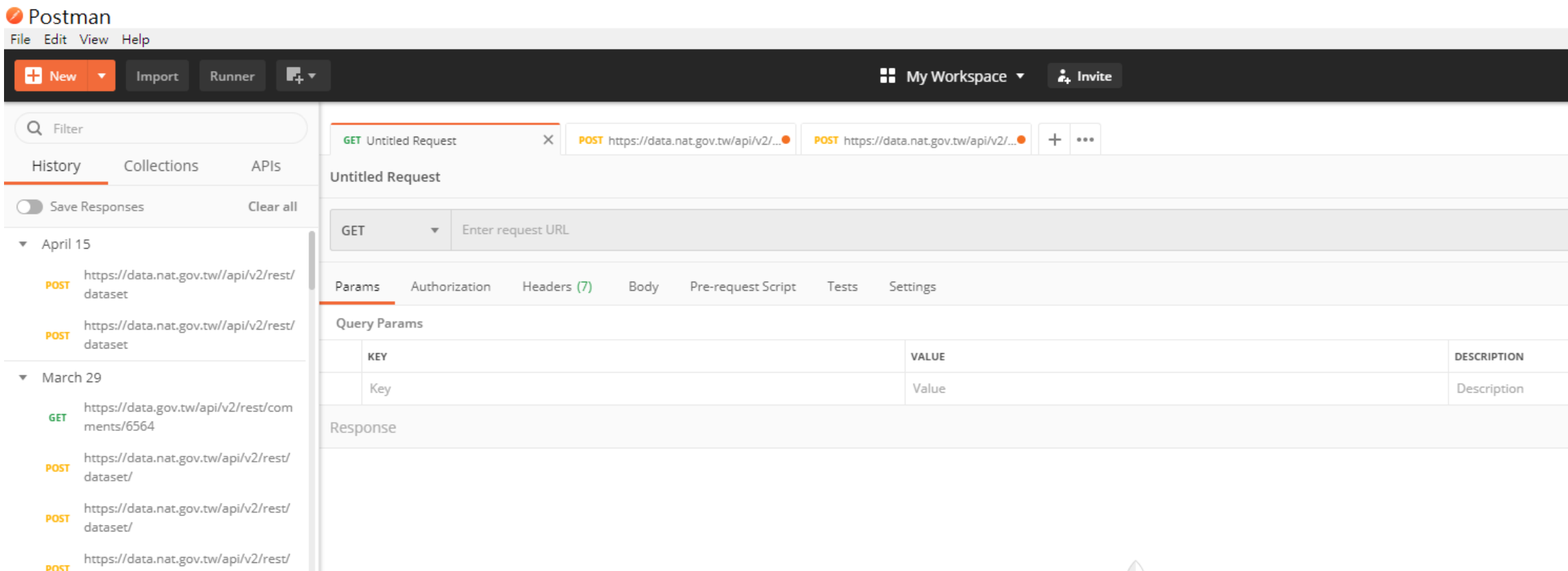


API介紹及應用

2) 應用程式介面(API) 相關工具介紹

2) API相關工具介紹(1)

- 透過POSTMAN工具可直接操作API程式。
- POSTMAN是一個可以模擬 HTTP Request 的工具。
 - 下載位置：<https://www.postman.com/downloads/>



2) API相關工具介紹(2)

The screenshot shows the Postman interface with a REST API request configured. The URL is `https://data.nat.gov.tw/api/v2/rest/dataset`. The method is set to `POST`. The body is in `JSON` format. The left sidebar shows a list of requests, including several `POST` requests to the same endpoint and a `GET` request to `https://data.gov.tw/api/v2/rest/comments/6564`.

REST動作 (REST Action): Indicated by the `POST` method selected in the interface.

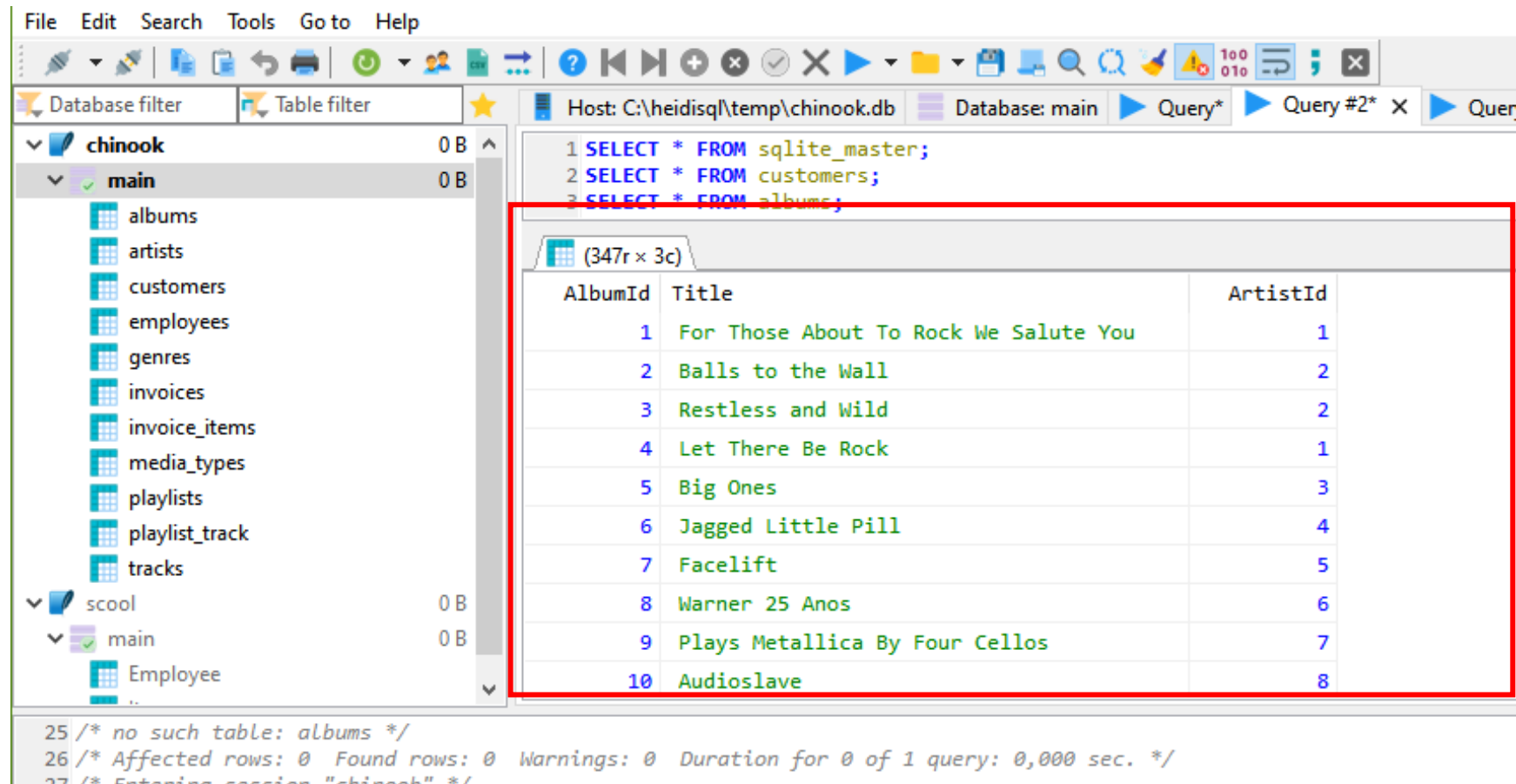
資源位置URI (Resource Location URI): Indicated by the URL `https://data.nat.gov.tw/api/v2/rest/dataset` in the request bar.

JSON格式內容 (JSON Format Content): Indicated by the JSON body content in the `Body` tab.

```
{
  "yTheme": "XXXXX",
  "yService": 400,
  "XXXXXX",
  "tion": "XXXXXX",
  "free",
  "erOID": "XXXX",
  "erContactName": "XXX",
  "erContactPhone": "XXXXX",
  "erContactEmail": "XXXXX",
  "requency": "XXXX",
  "requency": "XXXX",
  "edDate": "2020-04-09",
  "e": "XXXXX",
  "yDataset": "XXXX",
  "": "1",
  "vider": "XXXXX",
  "scription": "XXXXXXXXXX",
  "eStartedDate": "",
  "eEndedDate": "",
  "Coverage": "",
  "jUr1": "",
  "": "xxxxx"
}
```


2) API相關工具介紹(3)

- 客戶端要跟伺服器端索取資源，基本上等同與資料庫索取資源。
- 資料庫與Excel系列檔案格式概念類似，其中功能包含
 - 檢視資料
 - 新增資料
 - 更新資料
 - 刪除資料



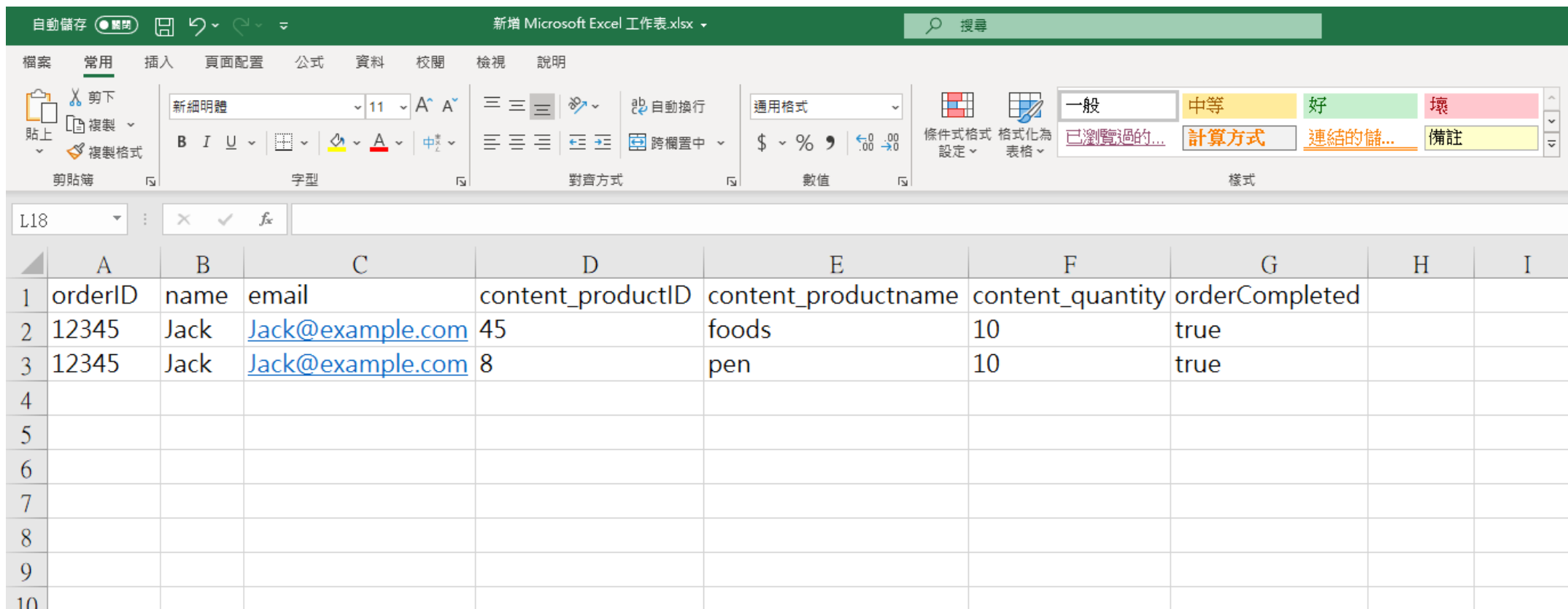
The screenshot shows a database client interface with a query window and a results grid. The query window contains three SQL statements: `1 SELECT * FROM sqlite_master;`, `2 SELECT * FROM customers;`, and `3 SELECT * FROM albums;`. The results grid displays the output of the third query, showing 10 rows of album data. The columns are AlbumId, Title, and ArtistId. The data is as follows:

AlbumId	Title	ArtistId
1	For Those About To Rock We Salute You	1
2	Balls to the Wall	2
3	Restless and Wild	2
4	Let There Be Rock	1
5	Big Ones	3
6	Jagged Little Pill	4
7	Facelift	5
8	Warner 25 Anos	6
9	Plays Metallica By Four Cellos	7
10	Audioslave	8

At the bottom of the interface, there is a status bar with the following text: `25 /* no such table: albums */`, `26 /* Affected rows: 0 Found rows: 0 Warnings: 0 Duration for 0 of 1 query: 0,000 sec. */`, and `27 /* Entering session "chinook" */`.

2) API相關工具介紹(4)

- 可以用Excel系列檔案格式想像成資料庫，都是以表格及欄位呈現。
- Excel可以使用排序、篩選功能檢視資料，直接於每一列或每一欄新增/更新/刪除資料。



The screenshot shows the Microsoft Excel interface with a data table. The table has the following structure:

	A	B	C	D	E	F	G	H	I
1	orderID	name	email	content_productID	content_productname	content_quantity	orderCompleted		
2	12345	Jack	Jack@example.com	45	foods	10	true		
3	12345	Jack	Jack@example.com	8	pen	10	true		
4									
5									
6									
7									
8									
9									
10									

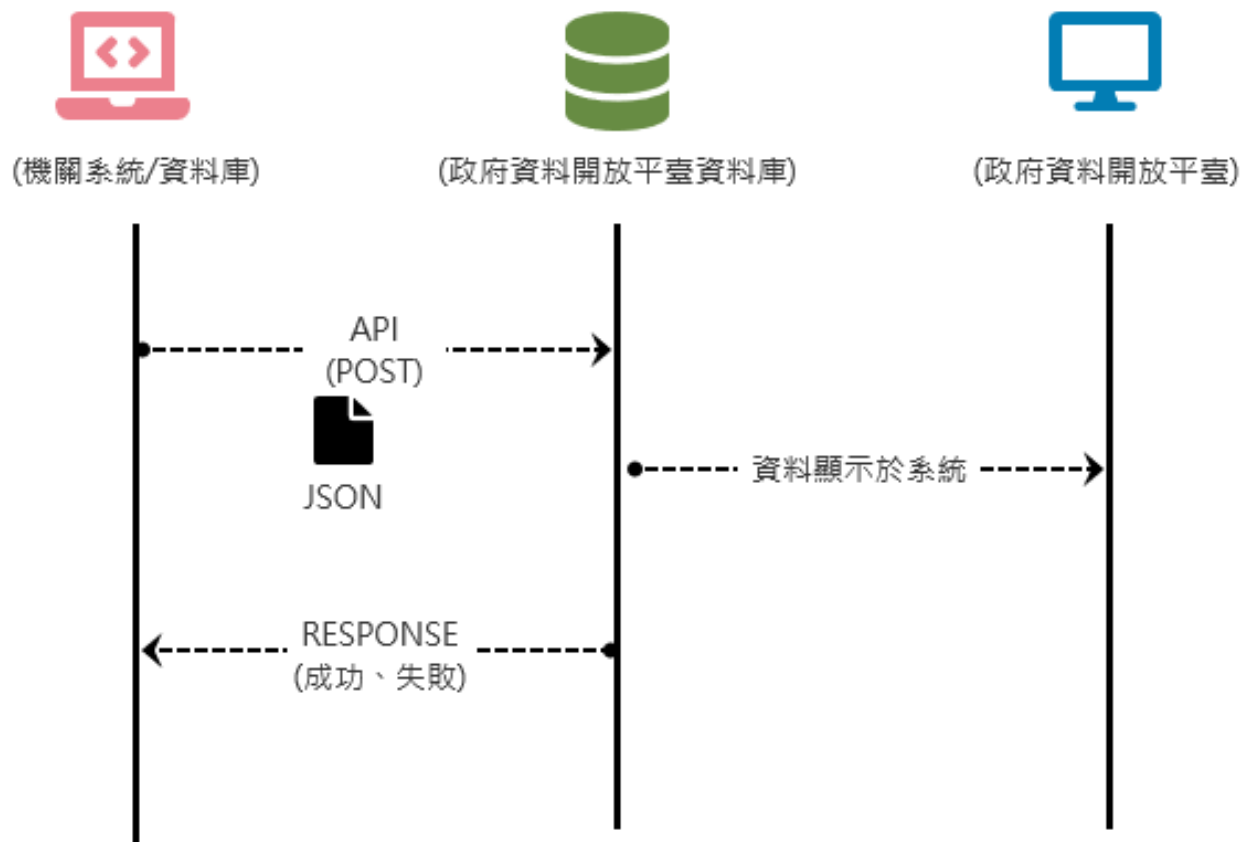


API介紹及應用

3) 應用程式介面(API) 應用

3) API實際操作的應用(1)

將機關資料庫之資料以API傳輸至
國發會政府資料開放平臺系統



機關想要將資料存放到政府資料開放平臺的範例說明。

3) API實際操作的應用-政府資料開放平臺(1)

1. 新增資料URI位置：

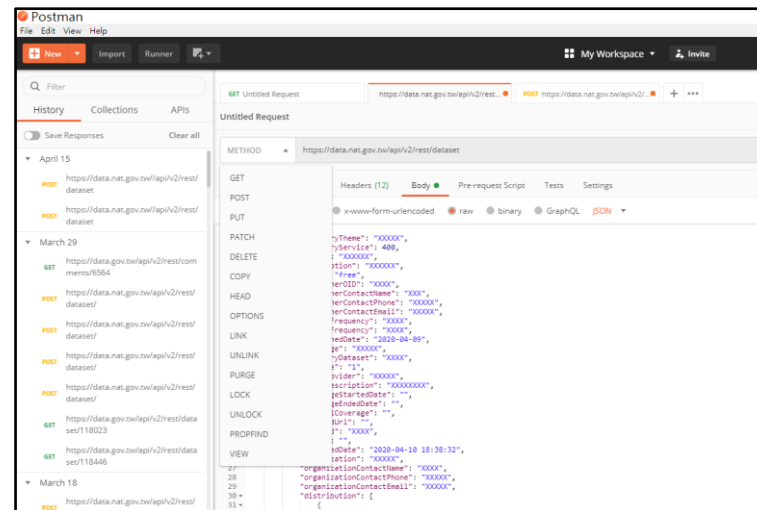
<https://data.gov.tw/api/v2/rest/dataset>

2. 新增資料動作：

POST

3. 新增資料JSON：

```
1 {  
2   "categoryTheme": "001",  
3   "categoryService": "I00",  
4   "categoryDataset": "A",  
5   "datasetId": "",  
6   "title": "政府資料開放平臺資料集清單",  
7   "description": "可下載開放資料清單",  
8   "license": "1",  
9   "cost": "free",  
10  "publisherOID": "國家發展委員會",  
11  "publisherContactName": "XXXXXX",  
12  "publisherContactPhone": "02-23165300",  
13  "publisherContactEmail": "XXXXXX@ndc.gov.tw",  
14  "dataProvider": "XXXXX",  
15  "publishedDate": "2013-11-01 08:00:00",  
16  "modifiedDate": "2020-03-04 09:48:42",  
17  "language": "zh",  
18 }  
19
```



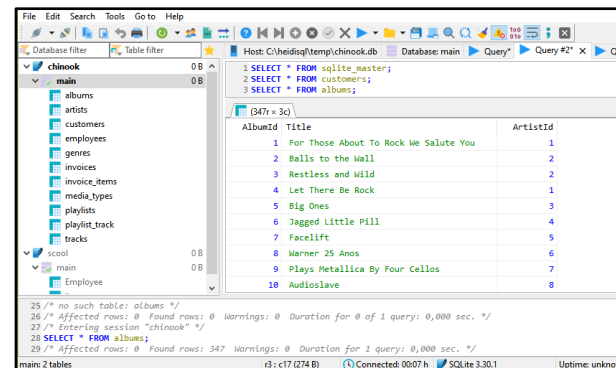
(POSTMAN工具/程式)

利用工具/程式將資料傳輸至政府資料開放平臺，或是取得政府資料開放平臺資料

3) API實際操作的應用-政府資料開放平臺(1)

資料集識別碼	資料集名稱	服務分類	主題分類	資料集分類	資料集描述	提供機關
27983	國家發展委員會性別平等專案小組委員名單	公共資訊	其他	開放資料	本會性別平等專案小	國家發展委員會
28468	國土空間發展策略計畫：國土空間結構全國階層	公共資訊	其他	開放資料	臺灣天然地形與地貌	國家發展委員會
28470	國土空間發展策略計畫：國土空間結構區域階層	公共資訊	其他	開放資料	據國際所形成城市區	國家發展委員會
28527	國家發展委員會圖書室外文圖書館藏目錄	公共資訊	其他	開放資料	提供國發會圖書室圖	國家發展委員會
28528	國家發展委員會圖書室中文圖書館藏目錄	公共資訊	其他	開放資料	提供國發會圖書室中	國家發展委員會
28592	中央行政機關室內公共區域免費手機充電站查詢服務	公共資訊	其他	開放資料	行政院及所屬各級機	國家發展委員會
28600	離島建設基金補助計畫清單	公共資訊	其他	開放資料	離島建設基金補助清	國家發展委員會

資料集識別碼	資料集名稱	服務分類	主題分類	資料集分類	資料集描述	提供機關
27983	國家發展委員會性別平等專案小組委員名單	公共資訊	其他	開放資料	本會性別平等專案小	國家發展委員會
28468	國土空間發展策略計畫：國土空間結構全國階層	公共資訊	其他	開放資料	臺灣天然地形與地貌	國家發展委員會
28470	國土空間發展策略計畫：國土空間結構區域階層	公共資訊	其他	開放資料	據國際所形成城市區	國家發展委員會
28527	國家發展委員會圖書室外文圖書館藏目錄	公共資訊	其他	開放資料	提供國發會圖書室圖	國家發展委員會
28528	國家發展委員會圖書室中文圖書館藏目錄	公共資訊	其他	開放資料	提供國發會圖書室中	國家發展委員會
28592	中央行政機關室內公共區域免費手機充電站查詢服務	公共資訊	其他	開放資料	行政院及所屬各級機	國家發展委員會
28600	離島建設基金補助計畫清單	公共資訊	其他	開放資料	離島建設基金補助清	國家發展委員會
6564	政府資料開放平臺資料集清單	公共資訊	其他	開放資料	可下載開放資料清單	國家發展委員會



(政府資料開放平臺資料庫)

新增一筆資料進資料庫，等同新增一列資料在Excel檔案裡的概念。

3) API實際操作的應用-政府資料開放平臺(1)

The screenshot shows the 'Government Data Open Platform Dataset Listing' page. At the top, there is a navigation bar with links like '全部資料集', '網站導覽', '互動專區', etc. Below the navigation bar, the breadcrumb trail reads '首頁 > 資料集 > 政府資料開放平臺資料集清單'. The main heading is '政府資料開放平臺資料集清單'. A '相關資料集' button is visible. The dataset details include: '資料集評分: 平均 3.4 (41 人次投票)', '資料集描述: 集中列示於政府資料開放平台資料集之詮釋資料...', '主要欄位說明: 資料集識別碼、資料集名稱、檔案格式、資料下載網址...', '資料下載網址: CSV (檢視資料 每日凌晨1時更新), XML (檢視資料 每日凌晨1時更新)', '提供機關: 國家發展委員會', '提供機關聯絡人姓名: 鄭先生 (0223165300#6853)', '更新頻率: 每日', '授權方式: 政府資料開放授權條款-第1版', '計費方式: 免費', '上架日期: 2013/11/01', '資料集類型: 原始資料', '詮釋資料更新時間: 2020/03/04 09:48', and '關鍵字: OpenData, 全部資料集, 本站清單'.

系統可將新增進資料庫的資料顯示於前臺。

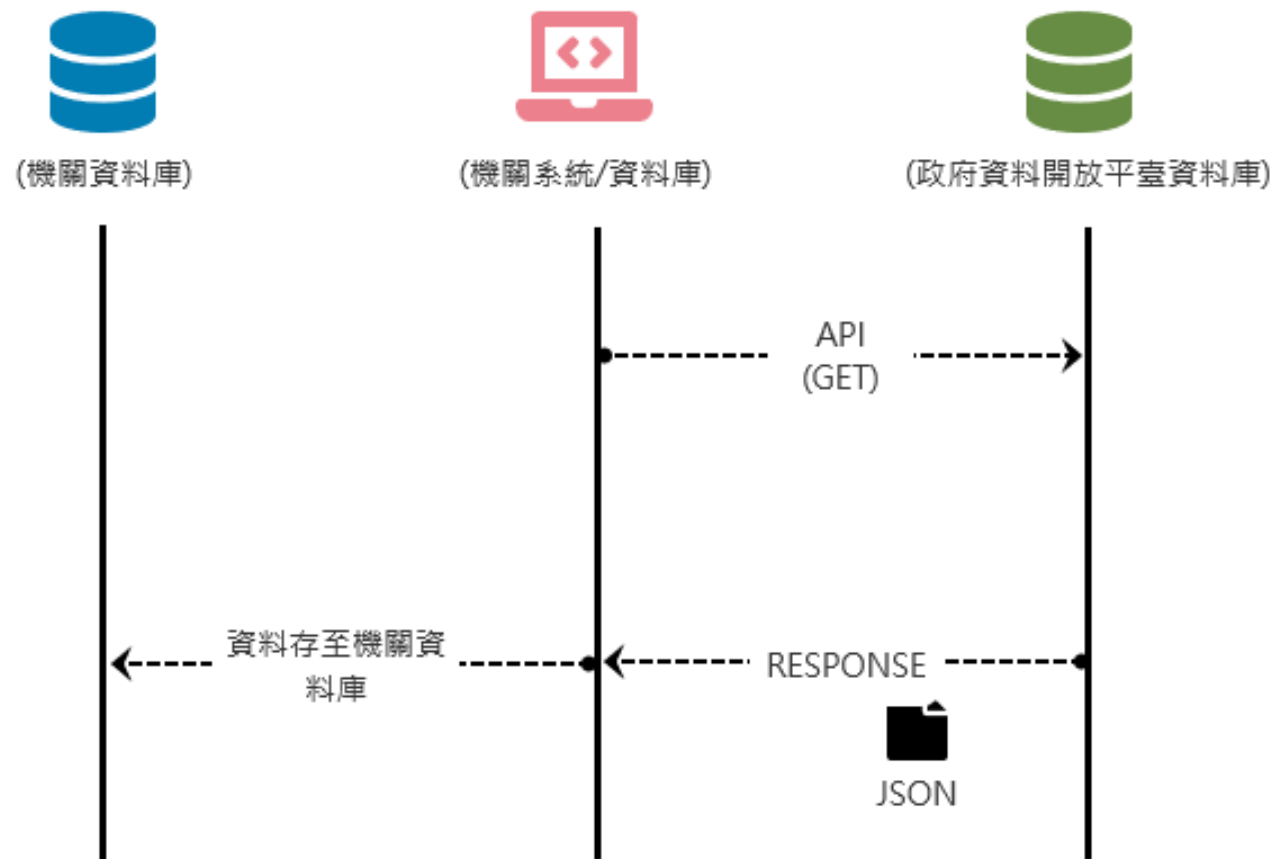
The screenshot shows the search results page on the Government Data Open Platform. It features a search bar at the top with the text '輸入您要查詢的關鍵字 (選填)'. Below the search bar, there are filters for '資料集數量: 44413', '資料集類型: 11553', '資料集格式: 220', and '資料集更新時間: 1100'. The main content area is titled '資料集列表 (Datasets)' and displays a list of datasets with columns for '資料集名稱', '資料集類型', and '資料集格式'. The list includes items like '行政院農業委員會農產產銷追溯系統', '107年度臺中市地方總決算', and '109年度各種商品抽驗結果'.

(政府資料開放平臺系統)

3) API實際操作的應用(2)

民眾/機關想要取得政府資料開放平臺的資料集/資料範例說明。

取得國發會政府資料開放平臺系統的資料



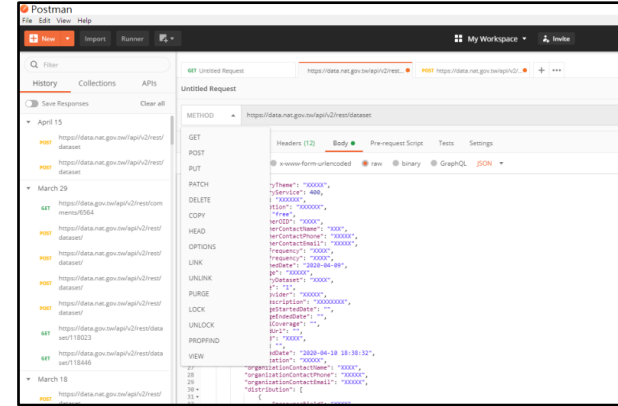
3) API實際操作的應用-政府資料開放平臺(2)

1. 取得資料URI位置：

<https://data.gov.tw/api/v2/rest/dataset/6564>

2. 取得資料動作：

GET



(POSTMAN工具/程式)

可利用工具/程式/瀏覽器(Chrome)將
URI位置貼上，與政府資料開放平臺
索取資料。

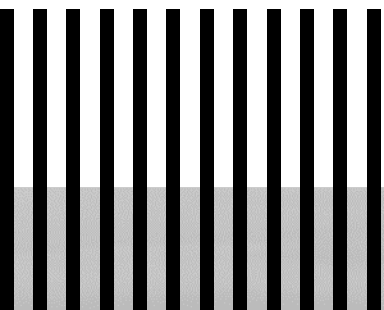
3) API實際操作的應用-政府資料開放平臺(2)

回傳一個JSON檔案給民眾
/機關。

```
1  {
2      "categoryTheme": "001",
3      "categoryService": "I00",
4      "categoryDataset": "A",
5      "datasetId": "",
6      "title": "政府資料開放平臺資料集清單",
7      "description": "可下載開放資料清單",
8      "license": "1",
9      "cost": "free",
10     "publisherOID": "國家發展委員會",
11     "publisherContactName": "XXXXXX",
12     "publisherContactPhone": "02-23165300",
13     "publisherContactEmail": "XXXXXX@endc.gov.tw",
14     "dataProvider": "XXXXX",
15     "publishedDate": "2013-11-01 08:00:00",
16     "modifiedDate": "2020-03-04 09:48:42",
17     "language": "zh",
18 }
19
20
```

API規範文件

#02





API規範文件檢視(1)

- 工程師撰寫完API後，必須產出API規範文件。
 - 使用者可透過文件直接使用API程式。
- API規範文件須包含**URI位置**、**動作**、**JSON**等內容資訊。
 - 可參考：<http://file.data.gov.tw/opendatafile/政府資料開放跨平臺介接規範第2版上稿版本.pdf>
- API規範文件須包含每個**資源欄位**的規範資訊。
 - 可參考：[https://file.data.gov.tw/content/01資料集詮釋資料標準規範第2版\(上稿版\)_20180903.pdf](https://file.data.gov.tw/content/01資料集詮釋資料標準規範第2版(上稿版)_20180903.pdf)

API規範文件檢視(2)

- API規範文件須包含**URI位置**、**動作**、**JSON**等內容資訊。

1. 新增詮釋資料

表-4 呼叫異動資料集詮釋資料之新增詮釋資料

功能說明	新增詮釋資料
服務路徑	{SRU}/api/v2/rest/dataset,Method=POST
輸入說明	輸入參數 以 JSON 格式示範資料集詮釋資料的欄位和內容： { "categoryTheme":"001", "categoryService":"I00", "categoryDataset":"A", "identifier":"A41000000G-000001", "title":"政府資料開放平臺資料集清單", "description":"集中列示於政府資料開放平臺資料集之詮釋資料式、資料集提供機關聯絡人、資料集提供機關聯絡人電話、備註等", "license":"1", "cost":"free", "dataProvider":"loginaccount" "publisherOID":"2.16.886.101.20003.20069.20001 國家發展委員會檔案管理局",

API規範文件檢視(3)

- API規範文件須包含每個資源欄位的規範資訊。

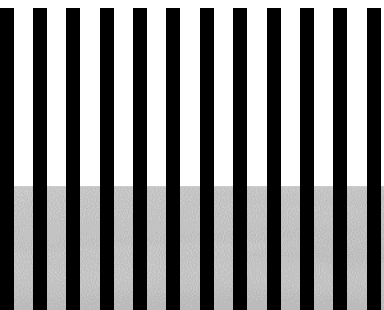
(一) 主題分類

表-6 主題分類

欄位名稱	主題分類
英文欄位名稱	categoryTheme
定義說明	主題分類編號 3 碼 6 類。
目的和意義	使用統一分類編號辭彙表有助於使用者搜尋資料資源。
選填條件	必須填寫
資料型態	列舉
填寫規則說明	共計 7 個分類項目，編碼及類別為： 001 其他 002 政府統計 003 政府預算 004 統一代碼 005 諮詢小組 006 政府支出 007 有限取用資料清單
建議範例	{"categoryTheme":"001"}
不建議範例	無

OAS介紹及驗證方式

#03





OAS介紹及驗證方式

1) OAS介紹



1-1) OAS目的

- 為實現共通性應用程式介面(Open API) , API 開發應符合：
 - OAS 標準之一致性描述方法
 - 提供機器可讀之標準格式 API 說明文件。
- 以「政府開放資料平臺」為例
- 導入國際 Open API Initiative (OAI) 組織之 OpenAPI Specification (OAS) 標準。
- 目的：提供各資料開放平臺使用者，以具有一致性的API取得資料
- 可大幅降低資料存取、API 調整以及維護等門檻，進而提升政府資訊效能

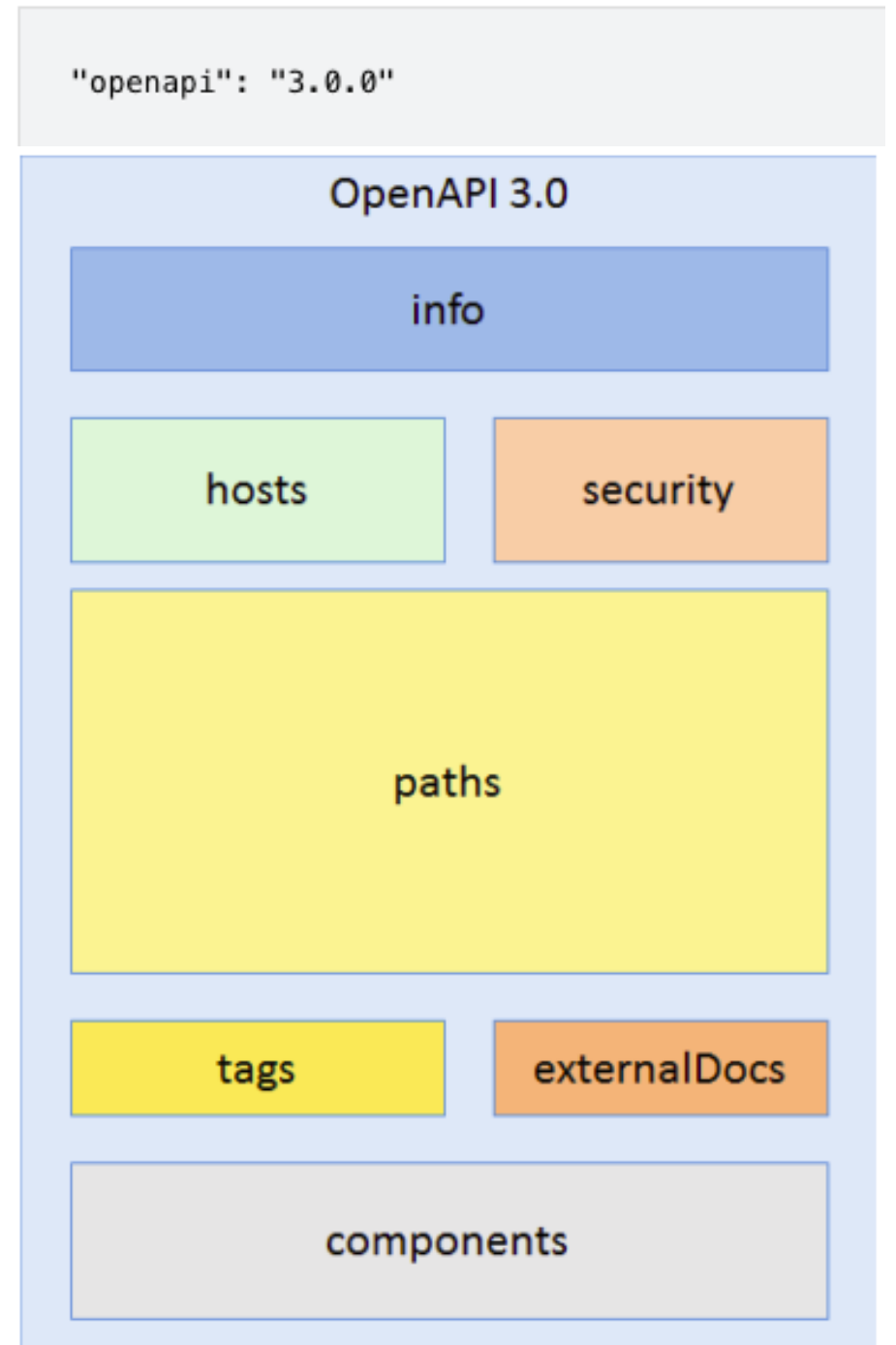


1-2) OAS介紹

- OAS是一個說明文件，可用 YAML(OAS3.0) 或 JSON 編寫。
- 該說明文件有固定欄位撰寫，其中包含必填及非必填欄位。
- OAS說明文件內之API需以RESTful API 的描述風格格式
- 相關資訊可參考：
 - <https://swagger.io/docs/specification/about/>
 - <http://spec.openapis.org/oas/v3.0.3>

1-3) OAS規範準則(1)

- OAS標準設計重點
 - OAS文件架構由根物件OpenAPI Object展開(其根物件內容包含info、hosts、security、paths等)。
 - 物件具備各自的固定欄位(Fixed Fields)，各欄位之類型(Type)則須符合基本類別或物件定義。
 - 詳細欄位內容及規範可參考：
<https://theme.ndc.gov.tw/lawout/Download.ashx?FileID=1439>



1-3) OAS規範準則(2)

- 根物件OpenAPI Object之欄位

欄位名稱	類型	描述
openapi	string	*必填 ，這字串必須載明該文件所使用的 OAS 版本
info	Info Object	*必填 ，提供這份 API 的詮釋資料
servers	[Server Object]	伺服器物件，可提供至目標伺服器之連結資訊。若未提供該欄位，或為空陣列，則伺服器欄位之預設 URL 將會是根目錄 "/"
paths	Paths Object	*必填 ，記載這份 API 的功能操作及可用路徑
components	Components Object	用於記載保存於各種 schema 之元素
security	[Security Requirement Object]	宣告其可跨用於整份 API 之安全機制 其清單內包含可供使用的 security requirement objects
tags	[Tag Object]	本標準於附加詮釋資料所使用的標籤清單。標籤順序可被分析工具所解析。清單中每個標籤名稱都必須是獨一無二的
externalDocs	External Documentation Object	附加的延伸文件

1-3) OAS規範準則(3)

- Info Object之欄位

欄位名稱	類型	描述
title	string	*必填 ，這個應用程式的標題
description	string	對此應用程式的簡短描述，可支援超文字格式
termsOfService	string	服務條款說明的網址
contact	Contact Object	這項 API 服務的公開聯絡資訊
license	License Object	這項 API 服務的公開授權方式
version	string	*必填 ，該 API 的版本

1-3) OAS規範準則(4)

OAS-JSON檔案樣式

```
{  
  "title": "Sample Pet Store App",  
  "description": "This is a sample server for a pet store.",  
  "termsOfService": "http://example.com/terms/",  
  "contact": {  
    "name": "API Support",  
    "url": "http://www.example.com/support",  
    "email": "support@example.com"  
  },  
  "license": {  
    "name": "Apache 2.0",  
    "url": "https://www.apache.org/licenses/LICENSE-2.0.html"  
  },  
  "version": "1.0.1"  
}
```

OAS-YAML檔案樣式

```
title: Sample Pet Store App  
description: This is a sample server for a pet store.  
termsOfService: http://example.com/terms/  
contact:  
  name: API Support  
  url: http://www.example.com/support  
  email: support@example.com  
license:  
  name: Apache 2.0  
  url: https://www.apache.org/licenses/LICENSE-2.0.html  
version: 1.0.1
```



1-4) 總結：

- 想像成OAS規範說明文件，代表為一個有規範目錄的WORD檔案。
- WORD檔案目錄包含openapi、info、server、paths等標題，其中openapi、info、paths標題為必填標題。
- Openapi標題內需填入這份文件的版本。
- Info標題內需要填入這份文件的名稱、描述等資訊。
- paths標題內需要填入API程式，但填入的API需要有一定的規範，例如API的網址規範、參數規範等。
- 因此，OAS文件才會稱為API規範的說明文件，欄位細節可參考：
<https://theme.ndc.gov.tw/lawout/Download.ashx?FileID=1439>



OAS介紹及驗證方式

2) OAS驗證方式



2) OAS驗證方式(1)

- 標準驗證方式

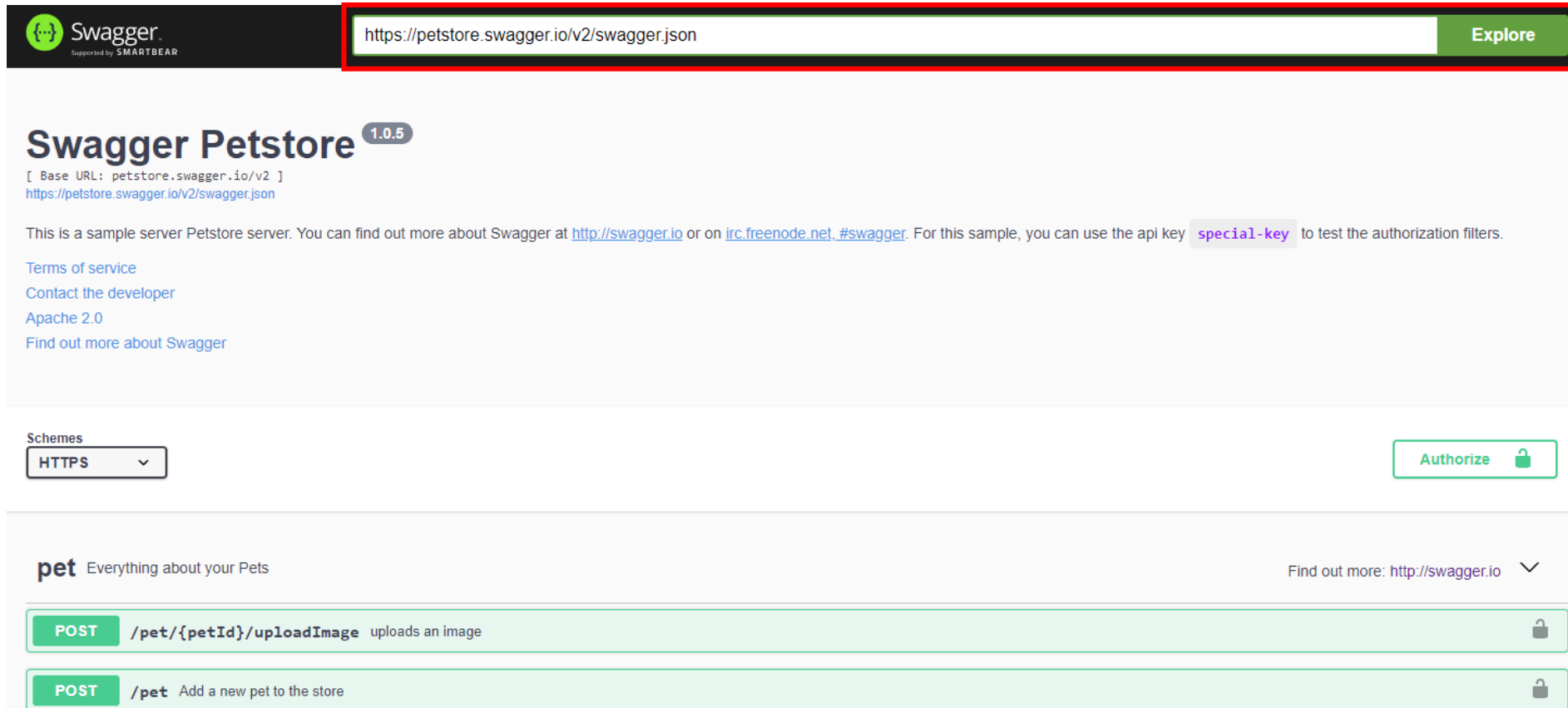
- 透過 swagger 所提供之驗證工具，對 OAS 文件進行檢測，網址：

<http://online.swagger.io/validator/debug?url=>

- 若OAS文件檢測結果無錯誤訊息且其顯示內容符合機關之專案需求，可視為通過驗證
- 透過swagger petstore進行驗證，網址：<https://petstore.swagger.io/>
 - 若OAS文件檢測結果通過則會顯示“ valid”訊息。
- 透過swagger editor進行驗證，網址：<https://editor.swagger.io/>
 - 將OAS文件丟至該網頁，則會呈現Restful API內容，若沒有呈現或有錯誤訊息顯示，則代表有欄位填寫方式非符合規範

2) OAS驗證方式(2)

- 透過swagger petstore進行驗證，網址：<https://petstore.swagger.io/>
 - 若OAS文件檢測結果通過則會顯示“ valid”訊息。



The screenshot shows the Swagger Petstore API interface. At the top, there is a search bar with the URL `https://petstore.swagger.io/v2/swagger.json` entered, highlighted with a red box. To the right of the search bar is an "Explore" button. Below the search bar, the page title is "Swagger Petstore" with a version number "1.0.5". The base URL is listed as `https://petstore.swagger.io/v2/swagger.json`. A paragraph of text explains that this is a sample server and provides instructions on how to use the API key `special-key` for authorization. There are links for "Terms of service", "Contact the developer", "Apache 2.0", and "Find out more about Swagger". Below this, there is a "Schemes" dropdown menu set to "HTTPS" and an "Authorize" button. The main content area shows the API endpoint `POST /pet/{petId}/uploadImage` with the description "uploads an image" and a lock icon. Below that, another endpoint `POST /pet` is shown with the description "Add a new pet to the store" and a lock icon.

2) OAS驗證方式(3)

- 透過swagger editor進行驗證，網址：<https://editor.swagger.io/>

The image shows the Swagger Editor interface. On the left, the Swagger JSON definition is displayed in a code editor. On the right, the rendered API documentation is shown, including the title 'Swagger Petstore 1.0.0', a description, and a list of endpoints.

```
1 swagger: "2.0"
2 info:
3   description: "This is a sample server Petstore server. You can find out more about Swagger at [http://swagger.io](http://swagger.io) or on [irc.freenode.net, #swagger](http://swagger.io/irc/). For this sample, you can use the api key `special-key` to test the authorization filters."
4   version: "1.0.0"
5   title: "Swagger Petstore"
6   termsOfService: "http://swagger.io/terms/"
7   contact:
8     email: "apiteam@swagger.io"
9   license:
10    name: "Apache 2.0"
11    url: "http://www.apache.org/licenses/LICENSE-2.0.html"
12 host: "petstore.swagger.io"
13 basePath: "/v2"
14 tags:
15 - name: "pet"
16   description: "Everything about your Pets"
17   externalDocs:
18     description: "Find out more"
19     url: "http://swagger.io"
20 - name: "store"
21   description: "Access to Petstore orders"
22 - name: "user"
23   description: "Operations about user"
24   externalDocs:
25     description: "Find out more about our store"
26     url: "http://swagger.io"
27 schemes:
28 - "https"
29 - "http"
30 paths:
31 /pet:
32   post:
33     tags:
34     - "pet"
35     summary: "Add a new pet to the store"
```

Swagger Petstore 1.0.0
[Base URL: petstore.swagger.io/v2]

This is a sample server Petstore server. You can find out more about Swagger at <http://swagger.io> or on <irc.freenode.net, #swagger>. For this sample, you can use the api key `special-key` to test the authorization filters.

[Terms of service](#)
[Contact the developer](#)
[Apache 2.0](#)
[Find out more about Swagger](#)

Schemes: **HTTPS**

pet Everything about your Pets [Find out more: http://swagger.io](http://swagger.io)

POST /pet Add a new pet to the store

PUT /pet Update an existing pet



OAS介紹及驗證方式

3) OAS檢測實例操作

3) OAS實例操作

進入Swagger editor，網址：
<https://editor.swagger.io/>

The image shows the Swagger Editor interface. On the left, a code editor displays a Swagger 2.0 API definition for a Petstore. The definition includes metadata like version (1.0.0), title, and contact information, as well as a list of tags and schemes. The right side shows the rendered preview of the API, including the title 'Swagger Petstore 1.0.0', the base URL, a description, and a list of links for terms of service, developer contact, and license. A dropdown menu for 'Schemes' is set to 'HTTPS'. At the bottom, a preview of the first API endpoint is shown: a POST request to '/pet' for adding a new pet to the store.

```
1 swagger: "2.0"
2 info:
3   description: "This is a sample server Petstore server. You can find out more about Swagger at [http://swagger.io](http://swagger.io) or on [irc.freenode.net, #swagger](http://swagger.io/irc/). For this sample, you can use the api key `special-key` to test the authorization filters."
4   version: "1.0.0"
5   title: "Swagger Petstore"
6   termsOfService: "http://swagger.io/terms/"
7   contact:
8     email: "apiteam@swagger.io"
9   license:
10    name: "Apache 2.0"
11    url: "http://www.apache.org/licenses/LICENSE-2.0.html"
12 host: "petstore.swagger.io"
13 basePath: "/v2"
14 tags:
15 - name: "pet"
16   description: "Everything about your Pets"
17 externalDocs:
18   description: "Find out more"
19   url: "http://swagger.io"
20 - name: "store"
21   description: "Access to Petstore orders"
22 - name: "user"
23   description: "Operations about user"
24 externalDocs:
25   description: "Find out more about our store"
26   url: "http://swagger.io"
27 schemes:
28 - "https"
29 - "http"
30 paths:
31 /pet:
```

Swagger Petstore 1.0.0
[Base URL: petstore.swagger.io/v2]

This is a sample server Petstore server. You can find out more about Swagger at <http://swagger.io> or on <http://swagger.io/irc/>. For this sample, you can use the api key `special-key` to test the authorization filters.

[Terms of service](#)
[Contact the developer](#)
[Apache 2.0](#)
[Find out more about Swagger](#)

Schemes
HTTPS

pet Everything about your Pets

POST /pet Add a new pet to the store

3) OAS實例操作

editor.swagger.io

Swagger Editor
Supported by SMARTBEAR

File Edit Generate Server Generate Client

- Import URL
- Import file
- Save as YAML
- Convert and save as JSON
- Clear editor

editor.swagger.io 顯示

Enter the URL to import from:

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

確定

取消

File>Import URL> 將OAS文件網址貼到輸入框中。

3) OAS實例操作

```
Swagger Editor
1 openapi: 3.0.0
2 info:
3   title: 國家發展委員會 政府資料開放平臺 我有話要說API Service
4   description: |
5     ## 國家發展委員會政府資料開放平臺提供資料使用者透過API Service取得我有話要說之詳細資訊。
6     * 我有話要說的標準規範：
7       我有話要說之資料規範可參考國家發展委員會所頒布的「[我有話要說、我想要更多、品質檢測介接規範](https://file
8       .data.gov.tw/content/%E5%9C%8B%E5%AE%B6%E7%99%BC%E5%B1%95%E5%A7%94%E5%93%A1%E6%9C%83_%E6%88%91%E6%9C%89%E8%A9
9       %B1%E8%A6%81%E8%AA%AA%E3%80%81%E6%88%91%E6%83%B3%E8%A6%81%E6%9B%B4%E5%A4%9A%E3%80%81%E5%93%81%E8%B3%AA%E6%AA
10      A2%E6%B8%ACAPI%E8%A6%8F%E7%AF%84%E6%96%87%E4%BB%B6_10903.pdf)」為原則。
11     * 我有話要說API 的使用： 我有話要說API
12       依據規範設計了兩個取得資訊，其一為政府資料開放平臺所有資料集底下的我有話要說編號，其二為針對某特定資料集底下
13       的我有話要說詳細資訊。
14   version: '2.0'
15 servers:
16   - url: 'https://data.gov.tw'
17 paths:
18   /api/v2/rest/comments:
19     get:
20       tags:
21       - 我有話要說API Service
22       description: 回傳政府資料開放平臺之所有資料集底下的我有話要說編號(CID)
23       parameters:
24       - name: limit
25         in: query
26         description: 搜尋資料集數量限制
27         allowEmptyValue: false
28         schema:
29           type: string
30           example: '10'
31       - name: order
32         in: query
33         description: 依據資料集編號排序(ASC是升序，DESC是降序)
34         allowEmptyValue: false
35         schema:
36           type: string
37           example: ASC
38 responses:
39   '200':
40     description: Definition generated from Swagger Inspector
41     content:
42       application/json:
43         examples:
44           '0':
```

呈現出兩欄，左欄為OAS文件的JSON格式內容，右欄為API的操作

國家發展委員會 政府資料開放平臺 我有話要說API Service 2.0 OAS3

國家發展委員會政府資料開放平臺提供資料使用者透過API Service取得我有話要說之詳細資訊。

- 我有話要說的標準規範： 我有話要說之資料規範可參考國家發展委員會所頒布的「[我有話要說、我想要更多、品質檢測介接規範](#)」為
- 我有話要說API 的使用： 我有話要說API 依據規範設計了兩個取得資訊，其一為政府資料開放平臺所有資料集底下的我有話要說編號為針對某特定資料集底下的我有話要說詳細資訊。

Servers
https://data.gov.tw

我有話要說API Service

GET /api/v2/rest/comments

GET /api/v2/rest/comment/dataset

3) OAS實例操作

```
Swagger Editor
Supported by SMARTBEAR

1 openapi: 3.0.0
2 info:
3   title: 國家發展委員會 政府資料開放平臺 我有話要說API Service
4   description: |
5     ## 國家發展委員會政府資料開放平臺提供資料使用者透過API Service取得我有話要說之詳細資訊。
6     * 我有話要說的標準規範：
7       我有話要說之資料規範可參考國家發展委員會所頒布的「[我有話要說、我想要更多、品質檢測介接規範](https://file
8         .data.gov.tw/content/%E5%9C%8B%E5%AE%B6%E7%99%BC%E5%B1%95%E5%A7%94%E5%93%A1%E6%9C%83_%E6%88%91%E6%9C%89%E8%A9
9         %B1%E8%A6%81%E8%AA%AA%E3%80%81%E6%88%91%E6%83%B3%E8%A6%81%E6%9B%B4%E5%A4%9A%E3%80%81%E5%93%81%E8%B3%AA%E6%AA%
10        A2%E6%B8%ACAPI%E8%A6%8F%E7%AF%84%E6%96%87%E4%BB%B6_10903.pdf)」為原則。
11      * 我有話要說API 的使用： 我有話要說API
12        依據規範設計了兩個取得資訊，其一為政府資料開放平臺所有資料集底下的我有話要說編號，其二為針對某特定資料集底下
13        的我有話要說詳細資訊。
14   version: '2.0'
15 servers:
16   - url: 'https://data.gov.tw'
17 paths:
18   /api/v2/rest/comments:
19     get:
20       tags:
21         - 我有話要說API Service
22       description: 回傳政府資料開放平臺之所有資料集底下的我有話要說編號(CID)
23       parameters:
24         - name: limit
25           in: query
26           description: 搜尋資料集數量限制
27           allowEmptyValue: false
28       schema:
```

左欄呈現JSON格式，依據規範列出Info Object、Paths等欄位。

3) OAS實例操作

國家發展委員會 政府資料開放平臺 我有話要說API Service ^{2.0} OAS3

國家發展委員會政府資料開放平臺提供資料使用者透過API Service取得我有話要說之詳細資訊。

- 我有話要說的標準規範：我有話要說之資料規範可參考國家發展委員會所頒布的「[我有話要說、我想要更多、品質檢測介接規範](#)」為原則。
- 我有話要說API 的使用：我有話要說API 依據規範設計了兩個取得資訊，其一為政府資料開放平臺所有資料集底下的我有話要說編號，其二為針對某特定資料集底下的我有話要說詳細資訊。

Servers

https://data.gov.tw

我有話要說API Service

GET /api/v2/rest/comments

GET /api/v2/rest/comment/dataset

右欄為API的操作，顯示出該API的URI位置、可以使用的動作等資訊

3) OAS實例操作

我有話要說API Service

GET /api/v2/rest/comments

回傳政府資料開放平臺之所有資料集底下的我有話要說編號(CID)

Parameters Try it out

Name	Description
limit string (query)	搜尋資料集數量限制
order string (query)	依據資料集編號排序(ASC是升序，DESC是降序)

Responses

將API展開，可輸入參數，並且點擊Try it out產出API撈取的資料。

3) OAS實例操作

在底下產出API之撈取結果。

Request URL

```
https://data.gov.tw/api/v2/rest/comments?limit=10&order=ASC
```

Server response

Code Details

200

Response body

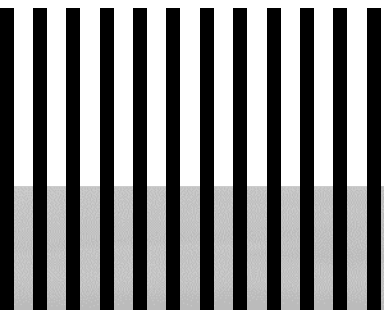
```
{
  "help": "",
  "success": true,
  "result": [
    {
      "datasetId": "5943",
      "title": "蒙藏補助團體私人情形季報表",
      "commentId": [
        "542313"
      ]
    },
    {
      "datasetId": "5944",
      "title": "蒙藏用語對照表",
      "commentId": [
        "9756"
      ]
    },
    {
      "datasetId": "5945",
      "title": "西藏統計資料",
      "commentId": [
        "10404"
      ]
    }
  ]
}
```

Download

Response headers

常見問答

#04



問題1：資料轉成API的好處？

答：

1. 固定檔案需定時更新資料內容，API可隨時撈取最新的資訊。
2. API可設計參數讓民眾使用。
3. 不須將資料存放至實體空間，只需存放至資料庫。
4. JSON易讓程式使用。

常見問答

火力發電溫室氣體排放量資料集有較高的下載量，但資料更新的方式為手動更新(目前資料停留至2018)，若可改成API撈取資料，則可給民眾最即時之資訊。

台灣電力公司_火力發電溫室氣體排放量

	A	B	C	D	E	F
1	年度	發電方式	排放量(萬噸)	占比(%)		
2	2015	燃油	792	9.28%		
3	2015	燃氣	2452	28.74%		
4	2015	燃煤	5289	61.98%		
5	2016	燃油	773	8.60%		
6	2016	燃氣	2591	28.60%		
7	2016	燃煤	5683	62.80%		
8	2017	燃油	847	8.47%		
9	2017	燃氣	2900	28.99%		
10	2017	燃煤	6255	62.54%		
11	2018	燃油	512	5.25%		
12	2018	燃氣	2889	29.66%		
13	2018	燃煤	6341	65.09%		
14						
15						
16						

相關資料集

資料集評分:

★★★★☆

平均 3.7 (11 人次投票)

資料集描述:

提供台電公司所屬火力發電溫室氣體排放量分析表

主要欄位說明:

年度、發電方式、排放量(萬噸)、占比(%)

資料下載網址:

CSV



檢視資料

發電方式，排放量(萬公噸CO2e)，占比.....

提供機關:

台灣電力股份有限公司

資料資源欄位: 年度、發電方式、排放量(萬噸)、占比(%)

檔案格式: CSV

編碼格式: UTF-8

資料量: 9

資料下載網址: http://data.taipower.com.tw/opendata/apply/file/d061002/台灣電力公司_火力發電_溫室氣體排放量.csv

資料資源描述: 發電方式，排放量(萬公噸CO2e)，占比(%), 以CSV表列

資料資源更新時間: 2020/02/18 16:45

詮釋資料更新時間:

2018/04/17 14:50

關鍵字:

主題分類:

其他

問題2：靜態資料(沒有系統資料庫)可否作成API?

答：

1. 固定的檔案格式也可以透過API程式抓取，但需要透過程式進行較多層次的轉換。
2. 固定檔案需以Excel系列、XML系列、以及JSON系列的檔案格式較易轉成API。

問題3：原本由同一資料庫產出不同資料集，改成API提供後，原資料集是否會異動(合併或保留不變)?

答：

1. 目前資料開放平臺內資料集的資料資源為固定檔案，若該資料集的資料資源改成API後還是需保留檔案，因為目前平臺無法提供存放API。
2. 可以待平臺開放API專區後，將資料集改成以API的方式提供，原來的資料集則可下架。

問題4：經由API存取資料，對資料庫而言是否容易產生資安問題？

答：

API程式在取得或存取資料與一般系統運行概念是一樣的，所以只要能在系統程式及資料庫方面做好環境資安的設定及檢查，是不會有風險問題產生。

問題5：API使用OAS規範的好處？

答：

1. 讓資料交換更容易使用。
2. 讓API更容易維護與調整。
3. 未來更易於導入Schema等資料標準。

問題6：哪一種資料集可以用API提供？哪種資料集無法？

答：

1. 每一個資料集都可以轉成API的方式提供。
2. 若資料集的資料是由系統產生/存在資料庫較易製作成API，若是非結構檔案格式(WORD、PPT、PDF等)較難做成API。

問題7：如果要開始將資料集的提供方式改為API，要怎麼挑選優先轉換的資料集？

答：

1. 資料集的資料是由系統產生/存在資料庫。
2. 資料集的資料檔案格式為結構化的資料(JSON、XML、CSV)，且通過品質檢測。
3. 資料集的使用率高(瀏覽、下載量高)的資料集。

問題8：說明從靜態資料到API的步驟及注意事項

答：

1. 步驟：將資料由系統產生並存放於資料庫→開發API程式(包含位置、動作(GET)、包成JSON檔案)→可使用POSTMAN進行測試→OAS 文件產出。
2. 注意事項：資料庫的資安環境要有規範、API程式製作需要思考是否要有權限制(例如:只開放給特定IP)、OAS可選擇最新版本去製作。



謝謝

對於操作方式有疑問

客服信箱：opendata@ndc.gov.tw

客服專線：0800-650-688 或 02-2531-1998