

共通性應用程式介面規範



大綱

- 壹、目的
- 貳、應用範圍
- 參、名詞定義
- 肆、規範準則
- 伍、範例實作
- 陸、常見問答

壹、目的

- 提供各資料開放平臺使用者以一致性應用程式介面 (Application Programming Interface, API) 取得資料
- 導入國際 Open API Initiative (OAI) 組織之 OpenAPI Specification (OAS) 標準
- 藉由一致性之描述方法，提供機器可讀之標準格式 API 說明文件，以大幅降低資料存取、API 調整以及維護等門檻，進而提升政府資訊效能

貳、應用範圍

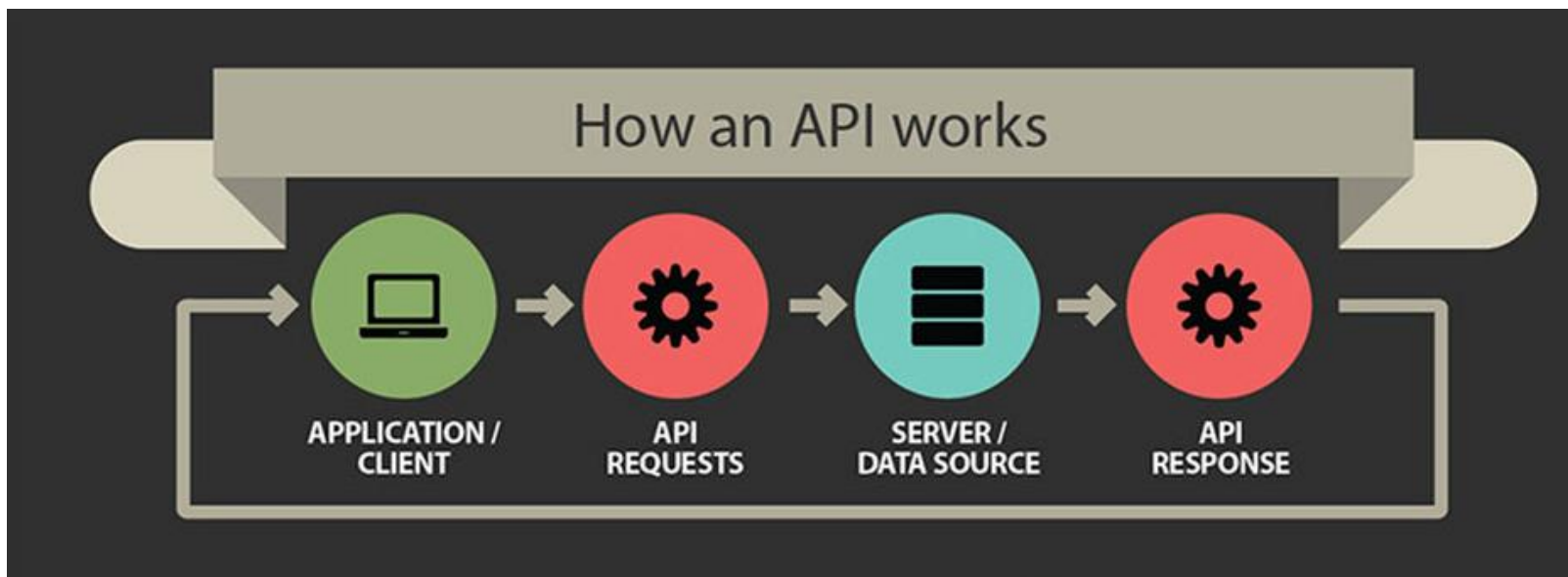
- 本規範適用於採 **API** 對外提供服務或其傳輸資料內容不涉及機敏性者，提供符合 **OAS** 標準之說明文件
- 不論該**API**所使用之授權及存取限制，均可採**OAS**標準進行描述

參、名詞定義

| 英文名稱 | 中文名稱 | 說明 |
|----------------|-----------------|--|
| API | 應用程式介面 | <u>A</u> pplication <u>P</u> rogramming <u>I</u> nterface 程式/系統/軟體 彼此相互銜接、溝通的約定方式 |
| REST / RESTful | 表現層狀態轉換 | <u>R</u> epresentational <u>S</u> tate <u>T</u> ransfer 基於 HTTP 架構上的一種全球資訊網路軟體建構風格 (請注意：RESTful 是設計風格、不是標準或守則) |
| JSON | JavaScript物件表示法 | <u>J</u> ava <u>S</u> cript <u>O</u> bject <u>N</u> otation 一個資料交換格式，各種不同程式語言皆可讀取/寫入 |
| YAML | -- | YAML Ain't Markup Language 以數據為中心的非標記型語言，（人類目視）可讀性比 JSON 來得高 |
| M2M | 機器對機器 | <u>M</u> achine to <u>M</u> achine |

快速複習：API

- API 是程式/系統/軟體 彼此相互銜接、溝通的約定方式
- 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



快速複習：REST / RESTful (1)

- 在 API 動作行為上，參考 HTTP request methods：
 - **GET**: 讀取資源 (safe & idempotent)
 - **PUT**: 替換資源 (idempotent)
 - **DELETE**: 刪除資源 (idempotent)
 - **POST**: 新增資源；也作為萬用動詞，處理其它要求
 - **PATCH**: 更新資源部份內容
 - **HEAD**: 類似GET，但只回傳HTTP header (safe & idempotent)
 - 完整版: https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol
 - **safe**：該操作不會改變伺服器端的資源狀態
 - **idempotent**：該操作不管做1遍或做n遍，都會得到同樣的資源狀態結果

快速複習： REST / RESTful (2)

- 在 API 資源命名原則上，參考 URI 命名規範：
 - 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}
 - 建議URI components都用小寫，兩個字之間用減號(dash) - 或底線
(underscore) _ 隔開皆可，但應保持一致

快速複習：REST / RESTful (3)

- 在 API 回傳結果，參考HTTP 回應狀態設計：
 - 2xx (成功)
 - 200 OK，API 成功無任何錯誤發生
 - 3xx (重新導向)
 - 在 RESTful API 上使用機會較低
 - 4xx (使用者端錯誤)
 - 401 使用者端未經過驗證
 - 403 使用者端被禁止使用
 - 404 請求的資源不存在
 - 5xx (伺服器端錯誤)
 - 500 伺服器端發生未知錯誤

快速複習：JSON

- 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，如：{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 (以 " " 包住)、數值 (number)、物件 (object)、布爾 (bool)、空 (null)、陣列 (array)
 - 例如：["string", 123, {key : value}, true, false, null, [1,2,3,4,5]]

```
01 {
02   "orderId": 12345,
03   "shopperName": "John Smith",
04   "shopperEmail": "johnsmith@example.com",
05   "contents": [
06     {
07       "productId": 34,
08       "productName": "SuperWidget",
09       "quantity": 1
10     },
11     {
12       "productId": 56,
13       "productName": "WonderWidget",
14       "quantity": 3
15     }
16   ],
17   "orderCompleted": true
18 }
```

快速複習：YAML (1)

- 以數據為中心的非標記型語言，可表達陣列、雜湊、純量等資料
- 以 `---` 作為檔案開頭，並用空白字元、換行字元、縮排來分隔資料
- 適合用來表達：各種資料結構、設定檔、檔案大綱
- 陣列用 `(- 空白字元)` 起始，或以 JSON 語法表示：

- 段落區塊：

```
---  
- 第一個項目  
- 第二個項目
```

- JSON相容格式：以 `(, 空白字元)` 區隔

```
---  
[第一個項目, 第二個項目, 第三個項目]
```

快速複習：YAML (2)

- 雜湊以 (: 空白字元)、換行符號區隔，或 JSON 相容格式：

- 段落區塊：

```
---  
name: 王小明  
age: 18
```

- JSON 相容格式：

```
---  
{name: 王小明, age: 18}
```

- 注意到 YAML 中的字串，不需要使用雙引號包覆

快速複習：YAML (3)

- 文字區段(多行文字)：
 - 可用 | 或 > 表示多行段落文字區段，兩者差異為：

```
---  
data: |  
    第一個項目  
    第二個項目  
data: >  
    第一個項目  
    第二個項目
```

- 兩者經過程式語言剖析後分別為：

```
{data: "第一個項目\n第二個項目\n", data: "第一個項目第二個項目\n"}
```

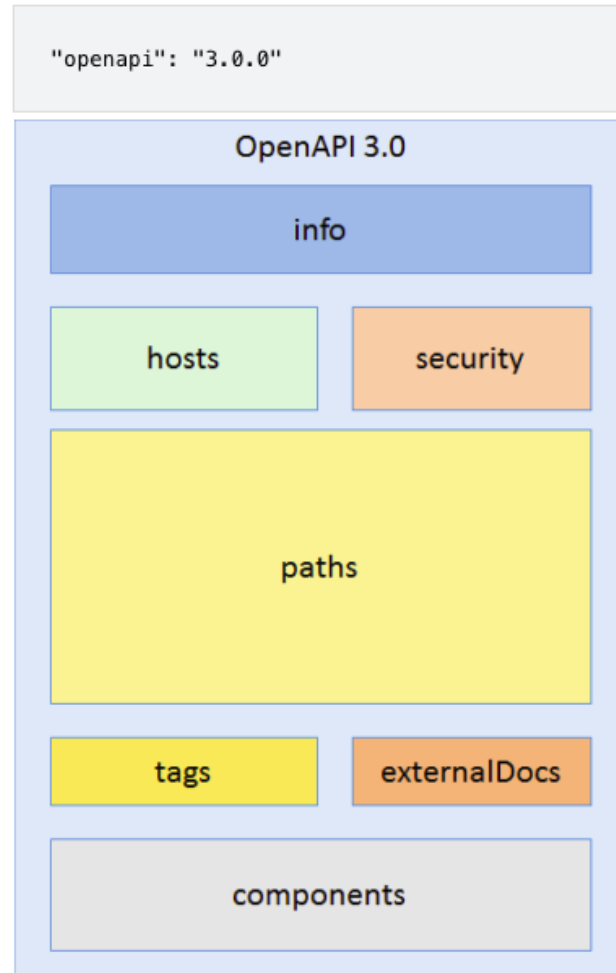
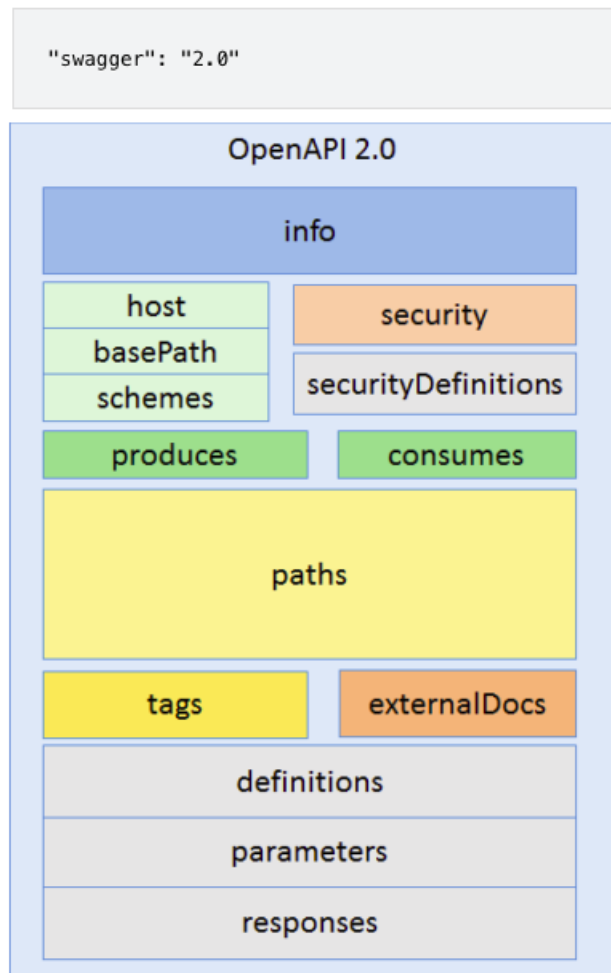
快速複習：OAS (1)

- 是一個 RESTful API 的描述格式
- OAS 可以描述整個 API，包括：
 - 每個端點上的可用端點（e.g., /users）和操作（e.g., GET/users）
 - 每個操作方法的輸入、和輸出參數
 - 驗證方法
 - 聯絡訊息、使用許可、使用條款和其他訊息
- 可用 YAML(OAS3.0) 或 JSON 編寫
- <https://swagger.io/docs/specification/about/>

快速複習：OAS (2)

- OAS2.0 / OAS3.0 的差別：
- <https://swagger.io/resources/open-api/>
- 結構定義略有不同
- OAS2.0 只支援 JSON
- OAS3.0 同時支援 YAML

快速複習：OAS (3)

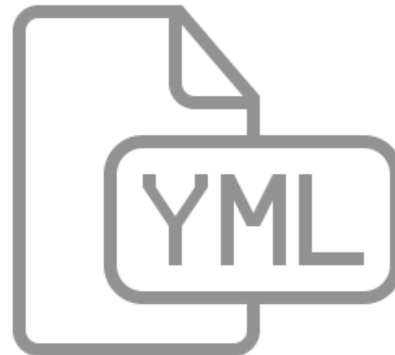
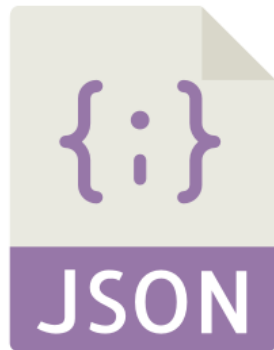


肆、規範準則 (1)

- 本規範之 API 設計原則：
 - 共通性
 - 採 Open API Initiative 組織之 OAS 標準，作為 API 說明文件之一致標準
 - <https://www.openapis.org/>
 - 輕便性
 - 採用 RESTful 風格的 API 呼叫方式
 - 標準化
 - 參考國際通用W3C相關標準及 OData.org 相關規範訂定之

肆、規範準則 (2)

- 一、符合OAS標準之API說明文件
 - (一) OAS標準設計重點
 - OAS文件需以JSON或YAML檔案格式呈現，並建議將命名為openapi.json或openapi.yaml。



肆、規範準則 (3)

- 一、符合OAS標準之API說明文件
 - (一) OAS標準設計重點
 - OAS文件架構由根物件OpenAPI Object展開
 - 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

```
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: http://www.apache.org/licenses/LICENSE-2.0.html
version: 1.0.1
```

肆、規範準則 (4)

| 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 |

肆、規範準則 (5)

| 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 |

肆、規範準則 (6)

```
{  
  "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"  
}
```

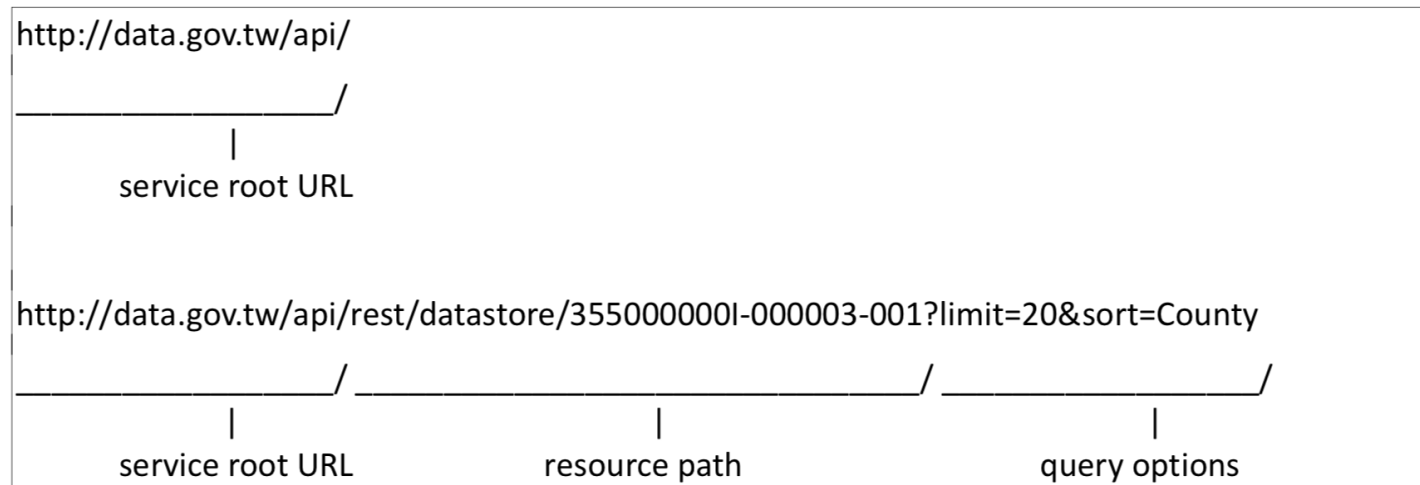
```
title: Sample Pet Store App  
description: This is a sample server for a pet store.  
termsOfService: http://example.com/terms/  
contact:  
  name: API Support  
  url: http://www.example.com/support  
  email: support@example.com  
license:  
  name: Apache 2.0  
  url: https://www.apache.org/licenses/LICENSE-2.0.html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

肆、規範準則 (7)

- 一、符合OAS標準之API說明文件
 - (二) OAS 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
 - <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

肆、規範準則 (8)

- 二、RESTful API 語法規則
 - Web API 採用 RESTful 設計方式
 - API 回傳內容以 JSON 為主
 - MIME type 為 application/json



肆、規範準則 (9)

- 三、API 版本描述方式
 - API 版本資訊應於 service root URL 說明
 - 使用 v1, v2, v3...(simple is best)
 - 不應使用 v-1.1, v1.2, 1.3 ...

```
http://data.gov.tw/api/v1/rest/datastore/355000000I-000003-001?limit=20&sort=County
```

service root URL

resource path

query options

伍、範例實作 (1)

- 一個常見的 API 說明文件差不多長這樣：

| | | | |
|------|-------------------|---|--|
| 功能說明 | | 取得資料集編號清單 | |
| 服務路徑 | | {SRU}/rest/dataset | |
| 輸入說明 | 參數名稱: modified | 參數輸入值 | 日期, 詮釋資料最後更新時間大於或等於輸入之查詢條件。若無輸入, 則傳回所有資料集編號清單。日期格式為 yyyy-MM-dd HH:mm:ss。 |
| | | 參數範例 | modified=2014-01-01 modified=2014-01-01%2023:59:59 |
| | 參數名稱: limit | 參數輸入值 | int, 限制最多回傳的資料筆數。 |
| | | 參數範例 | limit=1000 |
| | 參數名稱: offset | 參數輸入值 | int, 指定從第幾筆後開始回傳。可與 limit 配合使用達到分頁目的。 |
| | | 參數範例 | 每頁 10 筆、顯示第 4 頁如下 limit=10&offset=30 |
| 輸出說明 | 輸出內容 | 開放平臺上所有資料集之資料集編號。 | |
| | 輸出範例 | 以 JSON 格式示範取得資料集清單, 如下所示: ["A41000000G-000004", "301000000A-123456", "3550000001-000003", "3550000001-000004", "301000000A-000005"] 若查無資料, 則回傳: [] | |

伍、範例實作 (2)

- 一份符合OAS標準的API說明文件長這樣：

```
1 openapi: 3.0.0-rc2
2
3 servers:
4   - url: /api/v1
5
6 info:
7   version: '1.0'
8   title: 政府資料開放平台介面規範
9   description: >=
10    應用範圍、名詞定義、作業流程等請參照此[pdf檔案](http://file.data.gov.tw/opendatafile/%E6%94%BF%E5%BA%9C%E8%B3%)
11
12 paths:
13   /rest/dataset:
14     get:
15       operationId: listDatasetId
16       description: |
17         取得資料集編號清單。
18
19         * 輸出內容之資料集編號(identifier)係為資料集之資訊辨識編碼。
20
21         * 輸出內容之資料集編號等於資料集詮釋資料之資料集編號。
22       parameters:
23         - name: modified
24           in: query
25           description: |
26             日期, 詮釋資料最後更新時間大於或等於輸入之查詢條件。若無輸入, 則傳回所有資料集編號清單。
27             日期格式為 RFC3339 date-time。
28           required: false
29           schema:
30             type: string
31             format: date-time
32         - $ref: '#/components/parameters/limitParam'
33         - $ref: '#/components/parameters/offsetParam'
34       responses:
35         '200':
36           description: 資料開放平台所有資料集之資料集編號。若查無資料則回傳空陣列。
37           content:
38             application/json:
39               schema:
40                 type: object
41                 properties:
42                   success:
43                     type: boolean
44                     description: 表示回傳是否成功。
45                   result:
46                     type: array
47                     items:
48                       type: string
```

伍、範例實作 (3)

- 透過 **swagger** 工具後轉譯成視覺化文件如：

政府資料開放跨平臺介接規範 ^{1.0}

[Base url: /api/v1]
http://data.gov.tw/%E6%94%BF%E5%BA%9C%E8%B3%87%E6%96%99%E9%96%8B%E6%94%BE%E8%B7%A8%E5%B9%B3%E8%87%BA%E4%BB%8B%E6%8E%A5%E8%A6%8F%E7%AF%84_v2.0.yaml

應用範圍，名詞定義，作業流程等請參照此[pdf檔案](#)。

default

| | |
|--------|----------------------------|
| GET | /rest/dataset |
| POST | /rest/dataset |
| GET | /rest/dataset/{identifier} |
| PUT | /rest/dataset/{identifier} |
| DELETE | /rest/dataset/{identifier} |

伍、範例實作 (4)

- 一份通過驗證的OAS文件，可提供簡易地互動操作：

GET /rest/dataset

取得資料集編號清單。

- 輸出內容之資料集編號(identifier)係為資料集之資訊辨識編碼。
- 輸出內容之資料集編號等於資料集詮釋資料之資料集編號。

Parameters Cancel

| Name | Description |
|-------------------------------|---|
| modified string (query) | 日期, 詮釋資料最後更新時間大於或等於輸入之查詢條件。若無輸入, 則傳回所有資料集編號清單。日期格式為 RFC3339 date-time。 <input type="text" value="2017-06-01"/> |
| limit integer (query) | 限制最多回傳的資料筆數。 <input type="text" value="limit - 限制最多回傳的資料筆數。"/> |
| offset integer (query) | 指定從第幾筆後開始回傳。可與 limit 配合使用達到分頁目的。 <input type="text" value="offset - 指定從第幾筆後開始回傳。可與 limit 配合使用"/> |

Execute Clear

伍、範例實作 (5)

- 以「交通部公共運輸整合資訊流通服務平臺」為例
 - <http://ptx.transportdata.tw/MOTC/API/Main/docs/v2>

- OpenAPI 欄位：版本描述。

| JSON 格式 | YAML 格式 |
|--|-------------------------------|
| <pre>{ "openapi": "3.0.0-rc2",</pre> | <pre>openapi: 3.0.0-rc2</pre> |

- server 欄位：路徑描述。

| JSON 格式 | YAML 格式 |
|--|---|
| <pre>"servers": [{ "url": "http://ptx.transportdata.tw/MOTC" }],</pre> | <pre>servers: - url: http://ptx.transportdata.tw/MOTC</pre> |

伍、範例實作 (6)

- info 欄位：該 API 之基本詮釋資料。

| JSON 格式 | YAML 格式 |
|---|--|
| <pre>"info": { "version": "v2", "title": "MOTC Transport API V2", "description": "</div>\n <div class=\"info_description markdown\">資 料服務開發實作參考手冊: 請點我 公車動態使用注意事項:請點我 資料服務使用注意事項:請點 我 API URI Convention 文件說明:請點我< /a>" },</pre> | <pre>info: version: v2 title: MOTC Transport API V2 description: - </div> <div class="info_description markdown">資料服務開發實作參考手冊: 請點我 公車動態使用注意事項:請點我 資料服務使用注意事項:請點 我 API URI Convention 文件說明:請點我< /a></pre> |

伍、範例實作 (7)

- paths 欄位：該 API 各項功能之呼叫路徑，可與 server 欄位中的路徑結合為完整網址，並提供該項 API 之各項欄位、類別定義，以及範例說明。

(本案例僅摘要「取得指定[縣市]的公車動態定時資料(A1)」功能)

| JSON 格式 | YAML 格式 |
|---|---|
| <pre>"paths": { "/v2/Bus/RealTimeByFrequency/City/{City}": { "get": { "tags": ["CityBusApi"], "summary": "取得指定[縣市]的公車動態定時資料(A1)", "description": "市區公車之定時資料(A1)", "operationId": "CityBusApi_RealTimeByFrequency", "parameters": [{ "name": "City", "in": "path", "description": "欲查詢縣市", "required": true, "schema": { "type": "string", "enum": [{ "Text": "臺北市", "Value": "Taipei" }, { "Text": "新北市", "Value": "NewTaipei" }, { "Text": "桃園市", "Value": "Taoyuan" }, { "Text": "臺中市", "Value": "Taichung" }, { "Text": "臺南市", "Value": "Tainan" }] } }] } } }</pre> | <pre>paths: '/v2/Bus/RealTimeByFrequency/City/{City}': get: tags: - CityBusApi summary: '取得指定[縣市]的公車動態定時資料(A1)' description: 市區公車之定時資料(A1) operationId: CityBusApi_RealTimeByFrequency parameters: - name: City in: path description: 欲查詢縣市 required: true schema: type: string enum: - Text: 臺北市 Value: Taipei - Text: 新北市 Value: NewTaipei - Text: 桃園市 Value: Taoyuan - Text: 臺中市 Value: Taichung - Text: 臺南市 Value: Tainan - Text: 高雄市 Value: Kaohsiung - Text: 基隆市 Value: Keelung - Text: 新竹市 Value: Hsinchu - Text: 新竹縣 Value: HsinchuCounty - Text: 苗栗縣 Value: MiaoliCounty - Text: 彰化縣 Value: ChanghuaCounty</pre> |

伍、範例實作 (8)

- 利用 Swagger 將機讀格式轉換
 - <http://petstore.swagger.io/>

Swagger
Supported by SMARTBEAR

<http://ptx.transportdata.tw/MOTC/API/Main/docs/v2> Explore

MOTC Transport API V2 ^{v2}

[Base URL: ptx.transportdata.tw/MOTC]
<http://ptx.transportdata.tw/MOTC/API/Main/docs/v2>

本平臺提供涵蓋全國尺度之公車、臺鐵、高鐵、捷運、航空、自行車等公共運輸旅運資料服務API，歡迎各產政學單位介接使用。利用本平臺開放資料進行各項應用服務開發時，請考量不同特性使用者(如:性別/殘障/視障/老幼等)的需求，並歡迎回饋寶貴意見。

資料使用葵花寶典:[請點我](#)
資料服務開發實作參考手冊:[請點我](#)
API URI Convention文件說明:[請點我](#)

Schemes
HTTPS

AirApi

GET /v2/Air/Airport 取得機場資料

GET /v2/Air/Airport/{IATA} 取得指定[機場]資料

GET /v2/Air/Airline 取得航空公司資料

GET /v2/Rail/TRA/Network 取得臺鐵路網資料

取得臺鐵路網資料

Parameters Try it out

| Name | Description |
|--|--|
| \$select string (query) | 挑選 |
| \$filter string (query) | 過濾 |
| \$orderby string (query) | 排序 |
| \$top string (query) | 取前幾筆 Default value : 30 |
| \$skip string (query) | 跳過前幾筆 |
| \$format * required string (query) | 指定來源格式 Available values : JSON, XML |

伍、範例實作 (9)

- 也可以使用 Swagger Editor 線上編輯、測試
 - <https://editor.swagger.io/>

The screenshot shows the Swagger Editor interface. The left pane contains the Swagger JSON definition for the MOTC Transport API V2. The right pane displays the rendered API documentation, including the title "MOTC Transport API V2", the base URL, a description of the platform, and a list of API endpoints under the "AirApi" tag.

```
1 swagger: '2.0'
2 info:
3   version: v2
4   title: MOTC Transport API V2
5   description: |-
6     </div>
7     <div class="info_description markdown">
8       >本平臺提供涵蓋全國尺度之公車、臺鐵、高鐵、捷運、航空、自行車等公共運輸旅運資料服務API，歡迎
9       各產政學單位介接使用。<br><span class="swagger-description-indicatation">
10      >利用本平臺開放資料進行各項應用服務開發時，請考量不同特性使用者(如：性別/殘障/視障/老幼等
11      )的需求，並歡迎回饋寶貴意見。</span><br><br>資料使用葵花寶典:<a href=
12      https://ptxmtc
13      .gitbooks.io/ptx-api-documentation/content/ >請點我</a><br>資料服務開發實作參考手冊:<a href=
14      https://docs.google.com/viewer?url=https://github.com/ptxmtc/PTX_Web/blob/master
15      /Swagger%E6%9C%8D%E5%8B%99%E8%AA%AA%E6%98%8E%E4%B8%A%E5%82%B3%E5%8F%83%E8%80%83%E6%AA%94%E6%
16      A1%88/%E5%85%A%E5%85%B1%E9%81%8B%E8%B8%E6%95%B4%E5%90%88%E8%B3%87%E8%A8%8A%E5%B9%B3%E5%8F
17      %B0%E8%B3%87%E6%96%99%E6%9C%8D%E5%8B%99%E9%96%8B%E7%99%BCE5%AF%A6%E4%BD%9C.pdf?raw=true
18      >請點我</a><br>API URI Convention文件說明:<a href= https://docs.google.com/viewer?url=https
19      ://github.com/ptxmtc/PTX_Web/blob/master
20      /Swagger%E6%9C%8D%E5%8B%99%E8%AA%AA%E6%98%8E%E4%B8%A%E5%82%B3%E5%8F%83%E8%80%83%E6%AA%94%E6%
21      A1%88/API_URI_Convention%E6%96%87%E4%B8%B6_v1.pdf?raw=true >請點我</a>
22
23 host: ptx.transportdata.tw
24 basePath: /MOTC
25 schemes:
26   - http
27   - https
28 paths:
29   /v2/Air/Airport:
30     get:
31       tags:
32         - AirApi
33       summary: 取得機場資料
34       description: 取得所有機場資料
35       operationId: AirApi_Airport
36       consumes: []
```

MOTC Transport API V2 ^{v2}
[Base URL: ptx.transportdata.tw/MOTC]

本平臺提供涵蓋全國尺度之公車、臺鐵、高鐵、捷運、航空、自行車等公共運輸旅運資料服務API，歡迎各產政學單位介接使用。利用本平臺開放資料進行各項應用服務開發時，請考量不同特性使用者(如：性別/殘障/視障/老幼等)的需求，並歡迎回饋寶貴意見。

資料使用葵花寶典:[請點我](https://ptxmtc.gitbooks.io/ptx-api-documentation/content/)
資料服務開發實作參考手冊:[請點我](https://docs.google.com/viewer?url=https://github.com/ptxmtc/PTX_Web/blob/master/Swagger%E6%9C%8D%E5%8B%99%E8%AA%AA%E6%98%8E%E4%B8%A%E5%82%B3%E5%8F%83%E8%80%83%E6%AA%94%E6%A1%88/%E5%85%A%E5%85%B1%E9%81%8B%E8%B8%E6%95%B4%E5%90%88%E8%B3%87%E8%A8%8A%E5%B9%B3%E5%8F%B0%E8%B3%87%E6%96%99%E6%9C%8D%E5%8B%99%E9%96%8B%E7%99%BCE5%AF%A6%E4%BD%9C.pdf?raw=true)
API URI Convention文件說明:[請點我](https://docs.google.com/viewer?url=https://github.com/ptxmtc/PTX_Web/blob/master/Swagger%E6%9C%8D%E5%8B%99%E8%AA%AA%E6%98%8E%E4%B8%A%E5%82%B3%E5%8F%83%E8%80%83%E6%AA%94%E6%A1%88/API_URI_Convention%E6%96%87%E4%B8%B6_v1.pdf?raw=true)

Schemes
HTTP

AirApi

- GET /v2/Air/Airport 取得機場資料
- GET /v2/Air/Airport/{IATA} 取得指定[機場]資料
- GET /v2/Air/Airline 取得航空公司資料
- GET /v2/Air/Airline/{IATA} 取得指定[航空公司]資料

陸、常見問答

| 原則 | 效益 |
|-----------------|--------------------|
| 有建置/維護API就適用本規範 | 讓API更容易使用 |
| API說明文件需符合OAS標準 | 讓API更容易維護與調整 |
| 無需對已建置之API進行調整 | 避免政府資料交換重複開發 |
| 利用工具進行OAS標準之驗證 | 未來更易於導入Schema等資料標準 |

陸、常見問答

Q1: 若已經建置好的API是否仍應適用該規範？

Ans: 建議可於後續維護案中納入專案需求。

Q2: OAS的版本差異於履約期間應如何應對？

Ans: 履約版本建議可以招標文件時點之OAS版本為原則。

Q3: 撰擬/開發一份OAS文件需要的人力成本應如何估算？

Ans: 視API數量而定，軟體工程師通常可於1~2天內完成。