

# 共通性應用程式介面指引

數位發展部

中華民國111年12月

# 目 錄

壹、目的 .....	1
貳、應用範圍 .....	1
參、名詞定義 .....	1
肆、指引準則 .....	2
一、符合 OAS 標準之 API 說明文件 .....	3
(一) OAS 標準設計重點 .....	3
(二) OAS 標準驗證方式 .....	4
二、RESTful API 語法規則 .....	5
三、API 版本描述方式 .....	6
四、API Discovery .....	6
五、Open API 設計建議 .....	6
附錄 .....	7
附錄1—OpenAPI Specification 實際案例解析 .....	7
附錄2—相關網站連結 .....	12

## 壹、 目的

「共通性應用程式介面指引」目的為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, 以下簡稱 API) 取得資料。以 RESTful 風格為主要基礎，訂定應用程式介面的呼叫方式、語法規則及所提供的介面類型等項目，以達自動資料介接目標。

為擴大政府資訊服務效益，並保有各系統 API 開發彈性，爰導入國際 Open API Initiative 組織之 OpenAPI Specification (以下簡稱 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

本指引適用於可資料讀取/寫入之應用程式介面(API)，為使 API 具有共通性之特性，各機關採 API 對外提供服務或其傳輸資料內容不涉及機敏性之 API，請提供符合 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	定義
API (Application Programming Interface)	應用程式介面	為「『電腦作業系統 (Operating system)』或『程式函式庫』提供給應用程式呼叫使用的程式碼」。其主要目的是讓應用程式開發人員得以呼叫一組常式功能，而無須考慮其底層的原始碼為何、或理解其內部工作機制的細節。API 本身是抽象的，它僅定義了一個介面，而不涉及應用程式在實際實現過程中的具體操作。
REST	含狀態傳輸	全名為 Representational Stat

		e Transfer，是一種軟體架構設計風格。資源由 URI 指定，對資源的操作包括取得、創建、修改和刪除資源，這些操作正好對應 HTTP 協議提供之 GET、POST、PUT 和 DELETE 方法。
RESTful	含狀態傳輸的 Web 服務	是一個使用 HTTP 並遵循 REST 原則，以 URL 定位資源，根據 HTTP 內容指示操作動作與回應訊息。
JSON	—	一種常見的輕量級資料交換格式
YAML	—	一個可讀性高，用來表達資料序列的格式
SRU (Service Root URL)	服務根網址	描述平臺上提供各類別應用服務之網址。
Resource Path	資源路徑	接續於服務根網址後，指定某一資料集之資料資源路徑
Query Options	查詢選項	接續於資源路徑後，針對某一應用服務指定某一資料集之資源項目，表達所欲取得資料的範圍或查詢的條件
Metadata	詮釋資料	描述資料的資料
M2M (Machine to Machine)	機器與機器間資料交換	機器中的應用系統，已設定好定期呼叫機制與呼叫 API 方式，定期透過網際網路直接呼叫資料開放平臺提供之 API，以系統介接自動取得特定資料。

## 肆、 指引準則

本指引之 API 設計主要依循共通性、輕便性及標準化準則。

- 共通性：參採 Open API Initiative 組織之 OpenAPI Specification 標準，

作為 API 說明文件之一致標準。

- 輕便性：參考現階段及未來趨勢之 API 呼叫方式，採用 RESTful 風格 API。
- 標準化：參考國際通用 W3C 相關標準(如 URI、HTTP 等)及 OData.org 相關規範訂定之。

## 一、符合 OAS 標準之 API 說明文件

為實現共通性應用程式介面，API 開發應提供符合 OAS 標準<sup>1</sup>之一致性描述方法，提供機器可讀之標準格式 API 說明文件。

提供 OAS 實際案例解析如附錄1，並提供 OAS 標準之中文摘要譯本如附錄2供參。

### (一)OAS 標準設計重點

- 符合 OAS 標準之 API 說明文件(以下簡稱 OAS 文件)原則上由單一檔案製成。
- OAS 文件需以 JSON 或 YAML 檔案格式呈現 (API 本身的輸入和回傳值以及其他內容格式則未限制)，並建議將 API 說明文件命名為 openapi.json 或 openapi.yaml。
- OAS 文件之基本資料類型是依據 JSON Schema Specification Wright Draft 00 支援的類型訂定。其架構基於根物件 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。

以下摘要 OpenAPI Object 之固定欄位說明如下：

欄位名稱 Field Name	類型 Type	描述 Description
openapi	string	<b>必填</b> ，這字串必須載明該文件所使用的 OpenAPI Specification 版本。這個 openapi 欄位應該要讓使用者可藉由工具直譯其版本。該欄位與 API info 版本號無關。
info	Info Object	<b>必填</b> ，提供這份 API 的詮釋資料。如有需要，這詮釋

<sup>1</sup>OAI 官方正式發布 OAS 版本：<https://github.com/OAI/OpenAPI-Specification/tree/main/versions>

欄位名稱 Field Name	類型 Type	描述 Description
		資料亦可由使用者所使用。
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	<b>必填</b> ，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。僅需有一項 security requirement objects 滿足授權需求即可。可藉由個別操作覆蓋此定義。
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。並非所有 Operation Object 所使用的標籤都必須被宣告，未被宣告之標籤可被隨機地或邏輯性地組織起來。清單中每個標籤名稱都必得是獨一無二的。

## (二)OAS 標準驗證方式

API 開發人員可透過 Open API Initiative 所提供之官方驗證工具<sup>2</sup>，對 OAS 文件內容進行檢測，若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。

## 二、 RESTful API 語法規則

以政府資料開放平臺 (data.gov.tw) 為例，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，API 呼叫回傳內容格式則以 JSON 為主，若 API 輸出內容格式為 JSON，則 HTTP header Content-Type 為 application/json。

<sup>2</sup>OAI 提供之第三方驗證器，其驗證結果僅供參考：<https://github.com/OAI/OpenAPI-Specification/blob/main/IMPLEMENTATIONS.md>

服務路徑 URL 分為服務根網址(Service Root URL，簡稱 SRU)、資源路徑(Resource Path)和查詢選項(Query Options)：

- 服務根網址：平臺上提供該類別應用服務之網址。
- 資源路徑：接續於 SRU 後，以指定某一資源項目路徑名稱。
- 查詢選項：接續於資源路徑後，針對某一應用服務，指定所欲取得資料之範圍或查詢之條件。

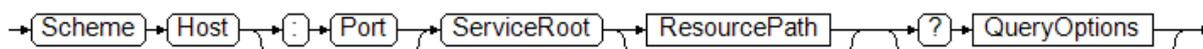


圖-1 URL 結構

依照上述 URL 結構定義，下圖以取得一資料集之資料資源內容 URL 為例。

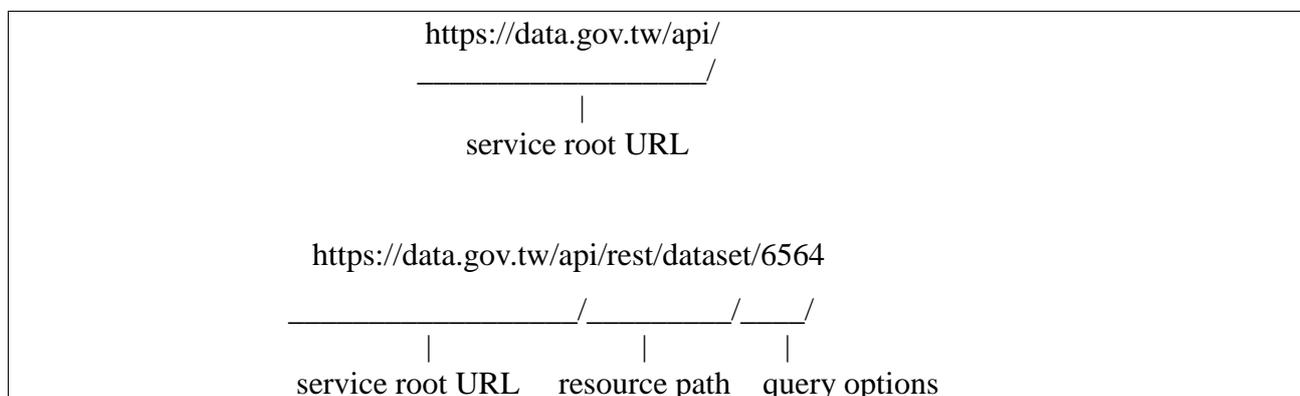


圖-2 URL 結構範例說明

### 三、API 版本描述方式

因 API 規格或實作可能有版本之演進，故如有需要描述版本資訊應於服務根網址(Service Root URL)說明，如下圖。版本資訊格式建議使用 v1、v2、v3，不建議使用 v-1.1、v1.2、1.3。

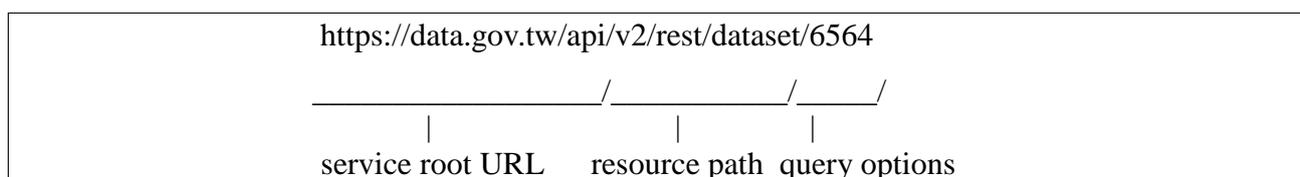


圖-3 含有版本資訊 URL 結構範例說明

## 四、 API Discovery

如機關所提供之 API 可供外界所查詢，建議可於首頁 html 文件或者是首頁的 Web 服務的 header 加入 API 鏈接關係，例如：

```
<link rel="api" type="application / apis + json" href="https://example.com/apis.json"/>
```

APIs.json 並非 OAS 標準的一部份，而是一個機器可讀的 sitemap 概念，可將該網站所有使用到的 API 進行列示。

## 五、 Open API 設計建議

- (一)建議高更新頻率資料或已有系統可即時產製之資料，可額外產製 Open API 供取用開放資料。
- (二)Open API 設計須符合上述 RESTful API 語法規則，並提供符合 OAS 3.0含以上版本規範之機器可讀說明文件。
- (三)依據「政府資料品質提升機制運作指引」，透過 Open API 取得之資料格式建議為 JSON、XML 格式，不建議回傳實體檔案或檔案下載連結。
- (四)若需身分驗證、會員註冊、API KEY 限制之 API，需於 OAS 說明文件內註明驗證方式，若該 API 無提供驗證資訊、需額外申請、需付費使用之情況，皆不屬於 Open API 範圍。



和 Client Secret 資訊並進行驗證，驗證完成後可開始於 Swagger UI 使用 API。

6. 欲透過程式介接 API，可參考[範例程式](<https://github.com/tdxmotc/SampleCode>)。

7. 為確保系統資源使用的合理分配與避免遭受濫用，於 Swagger UI 上使用 API 與程式介接 API 的行為將被記錄並定期做檢視。

##### API 呼叫次數限制:

1. 若不使用 API 金鑰呼叫 API，則僅能透過瀏覽器呼叫`【基礎】`服務，且每個呼叫來源端 IP 的上限為每日50次。

2. 使用 API 金鑰呼叫 API，每個呼叫來源端 IP 呼叫次數限制為50次/秒（無每日上限）。

API OAS 文本：[請點我](https://tdx.transportdata.tw/backend/api/File/Swagger/V3/2998e851-81d0-40f5-b26d-77e2f5ac4118)(<https://tdx.transportdata.tw/backend/api/File/Swagger/V3/2998e851-81d0-40f5-b26d-77e2f5ac4118>)",

```

"version": "v2"
},

```

金鑰】](</user/dataservice/key>)功能頁面，從預設金鑰(或建立新的金鑰)取得 Client Id 和 Client Secret 資訊。

5. 點選 Swagger UI 上的 Authorize 按鈕，依指示填入 Client Id 和 Client Secret 資訊並進行驗證，驗證完成後可開始於 Swagger UI 使用 API。

6. 欲透過程式介接 API，可參考[範例程式](<https://github.com/tdxmotc/SampleCode>)。

7. 為確保系統資源使用的合理分配與避免遭受濫用，於 Swagger UI 上使用 API 與程式介接 API 的行為將被記錄並定期做檢視。

##### API 呼叫次數限制:

1. 若不使用 API 金鑰呼叫 API，則僅能透過瀏覽器呼叫`【基礎】`服務，且每個呼叫來源端 IP 的上限為每日50次。

2. 使用 API 金鑰呼叫 API，每個呼叫來源端 IP 呼叫次數限制為50次/秒（無每日上限）。

API OAS 文本

:[請點我](https://tdx.transportdata.tw/backend/api/File/Swagger/V3/2998e851-81d0-40f5-b26d-77e2f5ac4118)(<https://tdx.transportdata.tw/backend/api/File/Swagger/V3/2998e851-81d0-40f5-b26d-77e2f5ac4118>)  
version: v2

- 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/Streaming/City/{City}": {     "get": {       "tags": [         "CityBus"       ],       "summary": "取得指定[縣市]的公車動態定時資料(A1)[逐筆更新]",       "description": "### 市區公車之定時資料(A1) ###\r\n- [逐筆更新]與[批次更新]之差異請詳見資料使用葵花寶典([連結](https://ptxmotc.gitbooks.io/ptx-api-documentation/content/api-zh-liao-shi-yong-zhu-yi-shi-xiang/buslive.html))",       "operationId": "CityBusApi_RealTimeByFrequency_UDP_2046",       "parameters": [         {           "name": "City",           "in": "path",           "description": "欲查詢縣市",           "required": true,           "schema": { </pre>	<pre> paths:   /v2/Bus/RealTimeByFrequency/Streaming/City/{City}:     get:       tags:         - CityBus       summary: 取得指定[縣市]的公車動態定時資料(A1)[逐筆更新]       description: "### 市區公車之定時資料(A1) ###\r\n- [逐筆更新]與[批次更新]之差異請詳見資料使用葵花寶典([連結](https://ptxmotc.gitbooks.io/ptx-api-documentation/content/api-zh-liao-shi-yong-zhu-yi-shi-xiang/buslive.html))"       operationId: CityBusApi_RealTimeByFrequency_UDP_2046       parameters:         - name: City           in: path           description: 欲查詢縣市           required: true           schema:             enum: </pre>

```

    "enum": [
      "Hsinchu",
      "HsinchuCounty",
      "MiaoliCounty",
      "ChanghuaCounty",
      "NantouCounty",
      "YunlinCounty",
      "ChiayiCounty",
      "Chiayi",
      "PingtungCounty",
      "YilanCounty",
      "HualienCounty",
      "TaitungCounty",
      "PenghuCounty",
      "Keelung"
    ],
    "type": "string"
  },
  "x-enum": {
    "Keelung": "基隆市",
    "YilanCounty": "宜蘭縣",
    "ChiayiCounty": "嘉義縣",
    "ChanghuaCounty": "彰化縣",
    "HsinchuCounty": "新竹縣",
    "MiaoliCounty": "苗栗縣",
    "PenghuCounty": "澎湖縣",
    "Hsinchu": "新竹市",
    "PingtungCounty": "屏東縣",
    "NantouCounty": "南投縣",
    "HualienCounty": "花蓮縣",
    "Chiayi": "嘉義市",
    "TaitungCounty": "臺東縣",
    "YunlinCounty": "雲林縣"
  }
},
{
  "name": "$select",
  "in": "query",
  "description": "挑選",
  "schema": {
    "type": "string"
  }
},
{
  "name": "$filter",
  "in": "query",
  "description": "過濾",
  "schema": {
    "type": "string"
  }
},
{
  "name": "$orderby",
  "in": "query",
  "description": "排序",
  "schema": {
    "type": "string"
  }
},
{
  "name": "$stop",

```

```

- Hsinchu
- HsinchuCounty
- MiaoliCounty
- ChanghuaCounty
- NantouCounty
- YunlinCounty
- ChiayiCounty
- Chiayi
- PingtungCounty
- YilanCounty
- HualienCounty
- TaitungCounty
- PenghuCounty
- Keelung
type: string
x-enum:
Keelung: 基隆市
YilanCounty: 宜蘭縣
ChiayiCounty: 嘉義縣
ChanghuaCounty: 彰化縣
HsinchuCounty: 新竹縣
MiaoliCounty: 苗栗縣
PenghuCounty: 澎湖縣
Hsinchu: 新竹市
PingtungCounty: 屏東縣
NantouCounty: 南投縣
HualienCounty: 花蓮縣
Chiayi: 嘉義市
TaitungCounty: 臺東縣
YunlinCounty: 雲林縣
- name: $select
in: query
description: 挑選
schema:
  type: string
- name: $filter
in: query
description: 過濾
schema:
  type: string
- name: $orderby
in: query
description: 排序
schema:
  type: string
- name: $stop
in: query
description: 取前幾筆
schema:
  type: integer
  default: 30
- name: $skip
in: query
description: 跳過前幾筆
schema:
  type: string
- name: health
in: query
description: 加入參數'?health=true'即可查詢此 API 服務的健康狀態

```

```

    "in": "query",
    "description": "取前幾筆",
    "schema": {
      "type": "integer",
      "default": 30
    }
  },
  {
    "name": "$skip",
    "in": "query",
    "description": "跳過前幾筆",
    "schema": {
      "type": "string"
    }
  },
  {
    "name": "health",
    "in": "query",
    "description": "加入參數'?health=true'即可查詢此 API 服務的健康狀態",
    "schema": {
      "enum": [
        "true",
        "false"
      ],
      "type": "string"
    }
  },
  {
    "name": "$format",
    "in": "query",
    "description": "指定來源格式",
    "required": true,
    "schema": {
      "enum": [
        "JSON",
        "XML"
      ],
      "type": "string"
    }
  }
],
"responses": {
  "200": {
    "description": "Success",
    "content": {
      "application/json": {
        "schema": {
          "title": "Array",
          "type": "array",
          "items": {
            "$ref": "#/components/schemas/PTX.Service.DTO.Bus.Specification.V2.BusA1Data"
          }
        }
      },
      "application/xml": {
        "schema": {
          "title": "Array",
          "type": "array",
          "items": {
            "$ref": "#/components/s

```

```

schema:
  enum:
    - 'true'
    - 'false'
  type: string
- name: $format
  in: query
  description: 指定來源格式
  required: true
  schema:
    enum:
      - JSON
      - XML
    type: string
responses:
  '200':
    description: Success
    content:
      application/json:
        schema:
          title: Array
          type: array
          items:
            $ref: >-
              #/components/schemas/PTX.Service.DTO.Bus.Specification.V2.BusA1Data
      application/xml:
        schema:
          title: Array
          type: array
          items:
            $ref: >-
              #/components/schemas/PTX.Service.DTO.Bus.Specification.V2.BusA1Data
  '299':
    description: 加入參數'?health=true'即可查詢此 API 服務的健康狀態
    content:
      application/json:
        schema:
          $ref: >- #/components/schemas/PTX.Service.DTO.Shared.Specification.V3.Base.DisplayHealth
      application/xml:
        schema:
          $ref: >- #/components/schemas/PTX.Service.DTO.Shared.Specification.V3.Base.DisplayHealth
  '304':
    description: >-
      服務端會在 Response 加上 Last-Modified
    header: 表示最近的更新時間。客戶端能利用此時間，於 Request 加上 If-Modified-Since
    header: 若沒有更新，服務端會回應304 StatusCode 且空值 Content
    content:
      application/json: {}

```

