



後端服務 API 規劃與共通 性指引適配說明

摘要

本文件旨在說明本專案後端 API 的設計方案，確保符合數位發展部「共通性應用程式介面指引」。文件中提供 YAML 檔案，並透過 SwaggerHub 提供清晰的 API 文件與互動式測試介面。

目錄

後端服務 API 規劃與共通性指引適配說明.....2

 一、OAS標準實踐.....2

 二、RESTful API 語法規則實踐.....6

 三、API版本描述6

後端服務 API 規劃與共通性指引|適配說明

本文件旨在說明本專案後端 API 的規劃符合全面符合數位發展部「共通性應用程式介面指引」的實際設計方案，提供YAML檔案並藉由 SwaggerHub 提供清晰的 API 文件與互動式測試介面。本專案後端API規劃與共通性應用程式介面指引之適配說明說明如下。

一、OAS標準實踐

1. 依據OAS規範，API 說明文件原則上由單一檔案製成，並且需以 JSON 或 YAML 檔案格式呈現。因此，本專案提供一YAML檔案作為API設計說明文件，檔名為 `openapi.yaml`。
2. OAS文件基本資料類型依據 JSON Schema Specification Wright Draft 00 支援的類型訂定，其相對應 OpenAPI Object 之固定欄位說明如下：

- (1) OpenAPI 欄位：用以載明該文件所使用的 OpenAPI Specification 版本。為必填欄位。本專案之設定資訊可詳見所提供之YAML檔案中之第1行，其設定值如下所示:

YAML格式
<code>openapi: 3.0.0</code>

- (2) info欄位：用以描述提供API之基本詮釋資料，為必填欄位。本專案之設定資訊可詳見所提供之YAML檔案中之第2~5行，其設定值如下所示:

YAML格式
<code>info: title: EIP API Test description: API endpoints for testing version: 1.0.0</code>

- (3) serve欄位：用以說明目標伺服器之連結資訊。其設定內容詳見所提供之YAML檔案中之第6~7行，目前所提供之設定值如下，其可根據服務的實際位置進行設定修改。

YAML格式
<code>servers: - url: https://modaeiptest.antnex.com.tw/modaeipapi/api/v1</code>

- (4) path欄位：用以描述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。其設定內容詳見所提供之YAML檔案中，各API之路徑設定。以下摘要「部內系統」區塊之功能。

YAML格式

```
paths:
  /applications:
    get:
      tags:
        - 部內系統
      summary: 查詢部內系統
      description: 返回所有部內系統
      responses:
        "200":
          description: Successful response
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/inline_response_200'
      security:
        - BearerAuth: []
    post:
      tags:
        - 部內系統
      summary: 新增部內系統
      requestBody:
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/applications_body'
        required: true
      responses:
        "200":
          description: Successful response
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/inline_response_200_1'
  /applications/{id}:
    get:
      tags:
        - 部內系統
      summary: 查詢部內系統
      description: 返回指定部內系統
      parameters:
        - name: id
          in: path
          description: applications id
          required: true
          style: simple
          explode: false
          schema:
            type: string
```

```

    example: af85141f-afa0-4556-8302-d0088dc6441c
  responses:
    "200":
      description: Successful response
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/inline_response_200_1'
      security:
        - BearerAuth: []
  put:
    tags:
      - 部內系統
    summary: 修改部內系統
    parameters:
      - name: id
        in: path
        description: application id
        required: true
        style: simple
        explode: false
        schema:
          type: string
          example: af85141f-afa0-4556-8302-d0088dc6441c
    requestBody:
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/applications_id_body'
      required: true
    responses:
      "200":
        description: Successful response
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/inline_response_200_1'

```

(5) components欄位：用於描述保存於各種schema之元素。在 components.schemas 中，每個 schema 通常對應資料庫中的一個資料表，表格的每個欄位則對應到該 schema 的屬性。其設定內容詳見所提供之YAML檔案中。以下摘要「部內系統」區塊之功能。

YAML格式

```

components:
  schemas:
    applications:
      required:
        - appId
        - code

```

```
- createdAtTime
- createdUserCode
- displayName
- id
- status
- updatedAtTime
- updatedUserCode
type: object
properties:
  id:
    type: string
    description: 物件識別碼
    format: uuid
  code:
    type: string
    description: 系統代碼
  displayName:
    type: string
    description: 系統名稱
  status:
    type: integer
    description: "狀態 - 1:啟用, 9:停用"
  createdapplicantCode:
    type: string
    description: 建立人員 - basic.users.userPrincipalName
  createdAtTime:
    type: string
    description: 建立時間
    format: date-time
  updatedUserCode:
    type: string
    description: 異動人員 - basic.users.userPrincipalName
  updatedAtTime:
    type: string
    description: 異動時間
    format: date-time
  appId:
    type: string
    description: 應用程式識別碼
    format: uuid
description: 部內系統
```

3. OAS標準驗證

本專案藉由Swagger所提供之檢測工具對所提供之openapi.yaml檔案進行檢測，檢測結果顯示本專案所提供之YAML檔案符合OAS規範。其檢測工具資訊與檢測結果顯示如下：

- 檢測工具：swagger-cil <https://github.com/APIDevTools/swagger-cli>
- 檢測結果：

```
~/Downloads > npx swagger-cli validate ./openapi.yaml 02:49:11 下午
./openapi.yaml is valid
```

二、RESTful API 語法規則實踐

1. 根據RESTful API規範，服務路徑URL分為服務根網址(Service Root URL，SRU)、資源路徑(Resource Path)和查詢選項(Query Options)。其中，服務根網址用以描述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。以本專案API服務規劃中「取得角色權限資料」交易為例，交易情境為查詢角色ID為1的權限資訊，依據規範說明如下：

- 「取得角色權限資料」交易URL路徑為：

<https://modaeiptest.antnex.com.tw/modaeipapi/api/v1/basic/rolefunction/1>

其中，

- 服務根網址(SRU)：https://modaeiptest.antnex.com.tw/modaeipapi/api/v1/
- 資源路徑(Resource Path)：basic/rolefunction/
- 查詢選項(Query Options)：1

2. HTTP方法的使用：本專案之API規劃設計遵循標準之HTTP方法，其應用說明如下，其規劃可詳見本轉案所提供之openapi.yaml文件。

- POST：用於建立新的資料
- GET：用於檢索資料集合
- PUT：用於更新完整的資料
- PATCH：用於更新資料的部分內容
- DELETE：用於刪除資料

三、API版本描述

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，且版本資訊格式建議使用 v1、v2、v3。本專案遵循此規劃，服務根網址(SRU)為https://modaeiptest.antnex.com.tw/modaeipapi/api/**v1**/，其中現行版本為v1。