

第 56 屆全國技能競賽-

競賽重點說明

職類名稱： 17 網頁技術

模組 D- Backend(RESTful API) 專輯管理系統

目錄

簡介	4
專案與任務說明	4
預設資料	4
公開 API	5
使用者 API	6
管理員 API	7
錯誤訊息	7
API	10
公開 API	10
使用者 API	10
管理員 API	10
API 資料標準	11
1.使用者登入	11
2.使用者註冊	11
3.取得所有專輯	11
4.取得專輯資訊	11
5.取得專輯封面圖片	11

6.取得專輯內歌曲	12
7.取得所有歌曲	12
8.取得歌曲封面圖片	12
9.使用者登出	12
10.取得歌曲訊息	12
11.取得統計結果	12
12.取得所有使用者	12
13.更新使用者角色	12
14.封鎖使用者	12
15.解除封鎖使用者	12
16.創建新專輯	13
17.更新專輯訊息	13
18.刪除專輯	13
19.新增歌曲到專輯	13
20.更新歌曲順序	13
21.更新歌曲訊息	13
22.自專輯刪除歌曲	13
23.取得專輯封面圖片連結	13
24.取得歌曲封面圖片連結	13
25.收藏歌曲	13
26.取消收藏歌曲	14
27.查看收藏清單	14
28.新增並套用標籤	14
29.移除標籤	14
評分摘要	14

簡介

本專案將建立一個專輯管理系統，用於協助辦公室管理者（以下稱為「管理員」）管理歌曲產品。歌曲產品屬於發行者，而發行者亦需管理其相關的專輯資料。管理員可檢視並管理這些發行者及其歌曲與專輯的列表及編輯內容。

此系統包含公開 API，包括：

- 使用者存取功能
- 專輯與歌曲的公開資訊功能
- 其他功能

專案與任務說明

本專案將建立一個專輯與歌曲產品管理系統，供發行者使用以管理其歌曲資料。

系統包含可用於查詢與讀取資料的 JSON API。

本系統應可透過 https://webXX.web.tw/XX_module_d/ 進行存取，其中 XX 代表崗位號碼。

預設資料

使用者資料表

Id	Username 使用者	email	Password 密碼	Role 角色
1	admin	admin@web.wsa	adminpass	admin
2	user1	user1@web.wsa	user1pass	user
3	user2	user2@web.wsa	user2pass	user

曲風標籤資料表

Id	曲風名
1	流行

2	搖滾
3	嘻哈
4	電子
5	爵士
6	經典
7	紓壓
8	鄉村

公開 API

任何使用者皆可透過公開 API 查詢所有專輯與歌曲資訊，使用此功能不需要註冊或登入。

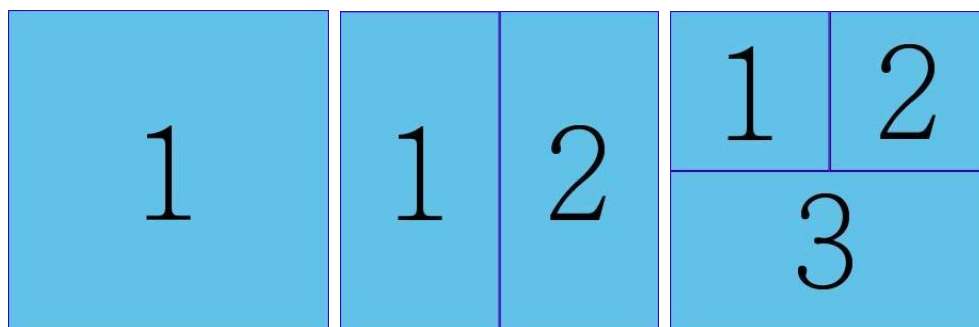
使用者可透過登入 API 進行登入。登入成功後，系統將提供一組 Access Token，用於存取需要授權的 API 功能。Access Token 由使用者的帳號進行 MD5 雜湊後轉換為全小寫十六進位字串產生。

當 API 需要分頁時，系統將採用游標式分頁。每次回應內容中會包含 next_cursor 欄位。使用者可將該值傳入下一次 API 請求，以取得下一頁資料；若 next_cursor 為 null，則表示已無更多資料可查詢。

next_cursor 的值為：取上一頁結果中最後一筆資料的唯一識別碼、將其包成 JSON，並以 Base64 進行編碼所得的字串。預設每次回傳 10 筆資料，使用者可透過 limit 參數調整每頁回傳的筆數。

舉例來說，若系統共有 ID 1~100 的紀錄，每頁回傳 10 筆，則第一次請求將回傳 ID 1~10 的資料，並於 next_cursor 回傳 Base64 編碼後的 {"id":10}。使用者於下一次請求傳入該 cursor 後，系統將回傳 ID 11~20 的資料，並依此類推，直到 next_cursor 為 null。

在查詢專輯列表 API 時，可使用 year 參數篩選指定年份的專輯。year 可輸入單一年份（如 2020），或年份區間（如 2018–2020）。若未傳入 year 參數，則回傳所有年份的專輯資料。



「取得專輯封面」的邏輯依照該專輯底下歌曲資料中的 `songs.is_cover` 屬性決定。最終封面為 1 或 2 或 3 張歌封面圖片的組合（如文件示例所示）。

封面圖片的組合順序需依照歌曲的顯示順序排列。若有超過 3 首歌曲被設定為封面，則需回傳 "Too many covers provided" 錯誤訊息。

當新增、修改、刪除或重新排列歌曲順序，且調整內容涉及 `is_cover` 狀態時，最終封面圖片的組合亦需更新。

例如，一張專輯包含五首歌曲：

- Song1 `is_cover:true`, `order:1`
- Song2 `is_cover:false`, `order:2`
- Song3 `is_cover:false`, `order:3`
- Song4 `is_cover:true`, `order:4`
- Song5 `is_cover:true`, `order:5`

此情況下，專輯封面需依序由歌曲 Song1、Song4 與 Song5 之封面組成共三張圖片。

使用者 API

使用者可透過登入 API 登入已註冊的帳號。當呼叫 API 時，Access Token 必須放置於 X-Authorization Header 中，API 會依據此 Token 進行使用者身份驗證。

當使用者登出時，系統必須刪除（或使其失效）該使用者的 Access Token，使該 Token 無法再次用於登入或存取授權資源。

若使用者在呼叫需要授權的 API 時未提供 Access Token，系統將回傳 401 錯誤狀態碼，並附上對應的錯誤訊息。

使用者可查詢歌曲詳細資訊。

使用者亦可取得統計結果，依照傳入的參數決定回傳內容：

?metrics=song: 回傳歌曲列表，並依照瀏覽數量由高到低排序。

?metrics=album: 回傳專輯列表，並依照總瀏覽數量由高到低排序。

?metrics=label: 回傳曲風分類，每項分類最多包含前 10 首瀏覽量最高的歌曲。

(選填) 可加入 labels 參數 (例如 : labels=Pop,Rock) 以篩選限定曲風。

管理員 API

當使用者角色為「admin」時，他們可管理其他使用者，包括查詢所有使用者、更新使用者資料，及封鎖 / 解除封鎖使用者。因此，被封鎖的使用者將無法存取任何使用者 API。

錯誤訊息

當發生錯誤時，系統必須依照情境回傳對應的錯誤訊息及 HTTP 狀態碼。

錯誤回傳格式：

Response (Unauthorized 401)	
Content-Type: application/json	
<pre>{ "success": false, "message": "Access Token is required" }</pre>	

錯誤訊息：

訊息	HTTP 狀態代碼	描述
Access Token is required 需要存取權杖	401	呼叫受保護 API 時，未在 X- Authorization 標頭中提供存取權杖。
Invalid Access Token 無效存取權杖	401	所提供的 Access Token 無效、已過期 或不存在
Access denied 存取遭拒	403	登入使用者嘗試操作不屬於自身的資 源 (例如：更新或刪除其他使用者的 專輯)。
Admin access required	403	非「admin」角色的使用者嘗試存取管

需要管理員存取權限		理員 API (例如 : GET /api/users) 。
User is banned 使用者遭封鎖	403	被封鎖的使用者試圖登入或存取受保護功能。
Not Found 不存在	404	請求的 API 路由不存在。 請求的資源不存在 (例如 : GET /api/albums/999) 。
Cover Not Found 封面圖片不存在	404	請求的專輯或歌曲封面圖片不存在。
Too many covers provided 太多封面圖片	400	該操作 (例如新增或修改歌曲) 導致專輯包含超過 3 張封面圖片。
Login failed 登入失敗	400	登入時提供了錯誤的帳號或密碼。
Username already taken 使用者名稱已被使用	409	註冊時提供的使用者名稱已被使用。
Email already taken Email 已被使用	409	註冊時提供的電子郵件地址已被使用。
Validation failed 認證失敗	400	請求中缺少必要欄位 (例如 : 註冊時未提供使用者名稱) 。 請求中的欄位格式不正確 (例如 : 電子郵件格式無效、release_year 不是數字) 。 更新歌曲順序時 , song_ids 包含無效 ID 或不屬於該專輯的歌曲 ID 。 嘗試將使用者角色更新為無效值。
Invalid parameter 無效參數	400	提供的查詢參數無效 , 例如 limit 超過 100 或不是數字。
Invalid cursor 無效游標	400	提供的游標格式錯誤或無效。

Invalid year format 無效年份格式	400	year 參數格式不正確 (例如 : "abc" 或 "2000-1990") 。
Invalid file type 無效檔案格式	400	上傳的 cover_image 格式不符合可接受的影像格式。
User not found 使用者不存在	404	管理員嘗試操作不存在的 user_id 。
Cannot ban self 不能封鎖自己	400	管理員嘗試封鎖自己的帳號。
Last admin demotion forbidden 禁止降級最後一位管理員	403	系統必須始終至少保留一位管理員。
Banned user update failed 無法更新被封鎖使用者	409	無法更新被封鎖使用者的角色。
Cannot ban another admin 無法封鎖另一位管理員	403	管理員無法封鎖另一位管理員。
Cannot delete default tag 不可刪除預設標籤	400	管理員不可刪除預設標籤。
Tag already taken 標籤名稱已被使用	409	標籤名稱已存在。

API

公開 API

	Item	Method	URL
1	User Login	POST	/api/login
2	User Register	POST	/api/register
3	Get All Albums	GET	/api/albums
4	Get Album Details	GET	/api/albums/{album_id}
5	Get Album Cover	GET	/api/albums/{album_id}/cover
6	Get Songs in Album	GET	/api/albums/{album_id}/songs
7	Get All Songs	GET	/api/songs
8	Get Song Cover	GET	/api/songs/{song_id}/cover
23	Get Album Cover Url	GET	/api/albums/{album_id}/cover_url
24	Get Song Cover Url	GET	/api/songs/{song_id}/cover_url

使用者 API

	Item	Method	URL
9	User Logout	POST	/api/logout
10	Get Song Details	GET	/api/songs/{song_id}
11	Get Statistics Result	GET	/api/statistics
25	Favorite song	POST	/api/songs/{song_id}/favorite
26	UnFavorite song	POST	/api/songs/{song_id}/unfavorite
27	Get Favorite songs	GET	/api/favorites

管理員 API

	Item	Method	URL
12	Get All Users	GET	/api/users
13	Update User Role	PUT	/api/users/{user_id}
14	Ban User	PUT	/api/users/{user_id}/ban
15	Unban User	PUT	/api/users/{user_id}/unban
16	Create New Album	POST	/api/albums
17	Update Album Details	PUT	/api/albums/{album_id}
18	Delete Album	DELETE	/api/albums/{album_id}
19	Add Song to Album	POST	/api/albums/{album_id}/songs
20	Update Song Order	PUT	/api/albums/{album_id}/songs/order
21	Update Song Details	POST	/api/albums/{album_id}/songs/{song_id}
22	Delete Song from Album	DELETE	/api/albums/{album_id}/songs/{song_id}
28	Create New Tag And Apply	POST	/api/tags
29	Remove Tag	DELETE	/api/tags/{tag_id}

API 資料標準

1. 使用者登入

POST /api/login
Request
Content-Type: application/json

```
{
  "username": "user1",
  "password": "user1pass"
}
```

Response (Success 200)

Content-Type: application/json

```
{
  "success": true,
  "data": {
    "token": "24c9e15e52afc47c225b757e7bee1f9d",
    "user": {
      "id": 1,
      "username": "user1",
      "email": "user1@web.wsa",
      "role": "user",
      "created_at": "2025-10-23T14:30:00.000Z",
      "updated_at": "2025-10-23T14:30:00.000Z"
    }
  }
}
```

2. 使用者註冊

POST /api/register

Request

Content-Type: application/json

```
{
  "username": "user4",
  "email": "user4@web.wsa",
  "password": "user4pass"
}
```

Response (Success 201 Created)

Content-Type: application/json

```
{
  "success": true,
  "data": {
    "user": {
      "id": 4,
```

```
    "username": "user4",
    "email": " user4@web.wsa",
    "role": "user",
    "created_at": "2025-10-23T15:00:00.000Z",
    "updated_at": "2025-10-23T15:00:00.000Z"
  }
}
```

3.取得所有專輯

GET /api/albums

Request

Query Parameters:

- capital: (string) e.g., "A"
- year: (string) e.g., "1980-2000"
- limit: (nubmer) e.g., 10
- cursor: (string) e.g., "eyJpZCI6IDEwfQ"

Example: /api/albums?filter=A&cursor=eyJpZCI6IDEwfQ

Response (Success 200)

Content-Type: application/json

```
{
  "success": true,
  "data": [
    {
      "id": 11,
      "title": "A Night at the Opera",
      "artist": "Queen",
      "release_year": 1975,
      "publisher": {
        "id": 1,
        "username": "user1",
        "email": "user1@web.wsa"
      }
    },
    {
      "id": 12,
      "title": "Abbey Road",
      "artist": "The Beatles",
      "release_year": 1969,
```

```
    "publisher": {
      "id": 2,
      "username": "user2",
      "email": "user2@web.wsa"
    }
  }
  ... // the rest of 8 items
],
"meta": {
  "prev_cursor": "e2lkOiAxMH0K",
  "next_cursor": "e2lkOiAyMX0K"
}
}
```

4. 取得專輯資訊

GET /api/albums/{album_id}

Request

Route Parameters:

- album_id: (integer) e.g., 12

Response (Success 200)

Content-Type: application/json

```
{
  "success": true,
  "data": {
    "id": 12,
    "title": "Abbey Road",
    "artist": "The Beatles",
    "release_year": 1969,
    "genre": "Rock",
    "description": "The eleventh studio album by the English rock band the Beatles.",
    "created_at": "2025-10-20T10:00:00.000Z",
    "updated_at": "2025-10-20T10:00:00.000Z",
    "publisher": {
      "id": 2,
      "username": "user2",
      "email": "user2@web.wsa"
    }
  }
}
```

5.取得專輯封面圖片

專輯封面將依照該專輯內歌曲的 `is_cover` 設定動態生成。

GET /api/albums/{album_id}/cover
Request Route Parameters: • <code>album_id</code>: (integer) e.g., 12
Response (Success 200) Content-Type: image/jpeg (Binary image data)

6.取得專輯內歌曲

回應的歌曲順序要依據設定的 `order` 欄位升冪排序

GET /api/albums/{album_id}/songs
Request Route Parameters: • <code>album_id</code>: (integer) e.g., 12
Response (Success 200) Content-Type: application/json <pre>{ "success": true, "data": [{ "id": 101, "album_id": 12, "title": "Come Together", "label":["a","b","c"], }] }</pre>

```

    "duration_seconds": 259,
    "order": 1,
    "is_cover": false,
    "cover_image_url": "/api/songs/101/cover"
  },
  {
    "id": 102,
    "album_id": 12,
    "title": "Something",
    "label": ["a", "b", "c"],
    "duration_seconds": 182,
    "order": 2,
    "is_cover": false,
    "cover_image_url": "/api/songs/102/cover"
  }
]
}

```

7. 取得所有歌曲

GET /api/songs

Request

Query Parameters:

- **keyword**: (string) e.g., "love"
- **limit**: (number) e.g., 10
- **cursor**: (string) e.g., "eyJpZCI6IDIwMH0"

Example: /api/songs?filter[keyword]=love

Response (Success 200)

Content-Type: application/json

```

{
  "success": true,
  "data": [
    {
      "id": 102,
      "album_id": 12,
      "title": "Something",
      "label": ["a", "b", "c"],
      "duration_seconds": 182,
      "album_title": "Abbey Road",
      "cover_image_url": "/api/songs/102/cover"
    }
  ]
}

```

```

    },
    {
      "id": 205,
      "album_id": 11,
      "title": "Love of My Life",
      "label": ["a", "b", "c"],
      "duration_seconds": 217,
      "album_title": "A Night at the Opera",
      "cover_image_url": "/api/songs/205/cover"
    }
  ],
  "meta": {
    "next_cursor": "eyJpZCI6IDlwNX0",
    "prev_cursor": "eyJpZCI6IDlwNH0="
  }
}

```

8. 取得歌曲封面圖片

GET /api/songs/{song_id}/cover
Request Route Parameters: song_id: (integer) e.g., 101
Response (Success 200) Content-Type: image/jpeg (Binary image data)

9. 使用者登出

POST /api/logout
Request X-Authorization: Bearer <token>

Response (Success 200)

Content-Type: application/json

```
{  
  "success": true  
}
```

10. 取得歌曲訊息

瀏覽次數將遞增。

GET /api/songs/{song_id}

Request

Route Parameters:

- `song_id`: (integer) e.g., 101

Response (Success 200)

Content-Type: application/json

```
{  
  "success": true,  
  "data": {  
    "id": 101,  
    "album_id": 12,  
    "title": "Come Together",  
    "duration_seconds": 259,  
    "order": 1,  
    "label": ["a", "b", "c"],  
    "view_count": 123,  
    "is_cover": false,  
    "lyrics": "Here come old flat top, he come grooving up slowly...",  
    "cover_image_url": "/api/songs/101/cover",  
    "created_at": "2025-10-20T10:01:00.000Z",  
    "updated_at": "2025-10-20T10:01:00.000Z"  
  }  
}
```

11.取得統計結果

若 metrics 為「label」或「album」，則以該項目進行分組，並依照 view_count 進行排序。
若 metrics 為「song」，則需依照一個或多個曲風（例如：Pop、Rock）進行過濾，並依照 view_count 排序。

GET /api/statistics
Request Query Parameters: - metrics: (string),(label album song) e.g., "label" - labels: (string),(optional) e.g., "Pop" Example: /api/statistics?metrics=song
Response (Success 200) Content-Type: application/json <pre>{ "success":true, "data":[{ "id": 101, "album_id": 12, "title": "Come Together", "duration_seconds": 259, "order": 1, "label":["a","b","c"], "view_count":123, "is_cover": false, "lyrics": "Here come old flat top, he come grooving up slowly...", "cover_image_url": "/api/songs/101/cover", "created_at": "2025-10-20T10:01:00.000Z", "updated_at": "2025-10-20T10:01:00.000Z" }, { "id": 100, "album_id": 10, "title": "Together", "duration_seconds": 248, "order": 1, "label":["a"], "view_count":12, "is_cover": false, "lyrics": "slowly...", "cover_image_url": "/api/songs/100/cover", "created_at": "2025-10-20T10:01:00.000Z", "updated_at": "2025-10-20T10:01:00.000Z" }] }</pre>

```
    },  
  ],  
}
```

Example: /api/statistics?metrics=album

```
{  
  "success":true,  
  "data":[  
    {  
      "id": 13,  
      "title": "My Updated Album Title",  
      "artist": "My Band",  
      "release_year": 2025,  
      "genre": "Indie Rock",  
      "description": "Updated description.",  
      "publisher": {  
        "id": 1,  
        "username": "admin",  
        "email": "admin@web.wsa"  
      },  
      "created_at": "2025-10-23T16:00:00.000Z",  
      "updated_at": "2025-10-23T16:05:00.000Z",  
      "total_view_count": 17  
    },  
    {  
      "id": 12,  
      "title": "My Updated Album Title 2",  
      "artist": "My Band 2",  
      "release_year": 2021,  
      "genre": "Indie Rock",  
      "description": "Updated description.",  
      "publisher": {  
        "id": 1,  
        "username": "admin",  
        "email": "admin@web.wsa"  
      },  
      "created_at": "2025-10-23T16:00:00.000Z",  
      "updated_at": "2025-10-23T16:05:00.000Z",  
      "total_view_count": 10  
    },  
  ],  
}
```

Example: /api/statistics?metrics=label

```
{  
  "success":true,  
  "data":{  
    {  
      "total_view_count": 6,  
      "label": "Rock",  
      "songs": [  
        {  
          "id": 1,  
          "title": "My Updated Album Title",  
          "artist": "My Band",  
          "release_year": 2025,  
          "genre": "Indie Rock",  
          "description": "Updated description.",  
          "publisher": {  
            "id": 1,  
            "username": "admin",  
            "email": "admin@web.wsa"  
          },  
          "created_at": "2025-10-23T16:00:00.000Z",  
          "updated_at": "2025-10-23T16:05:00.000Z",  
          "total_view_count": 17  
        },  
        {  
          "id": 12,  
          "title": "My Updated Album Title 2",  
          "artist": "My Band 2",  
          "release_year": 2021,  
          "genre": "Indie Rock",  
          "description": "Updated description.",  
          "publisher": {  
            "id": 1,  
            "username": "admin",  
            "email": "admin@web.wsa"  
          },  
          "created_at": "2025-10-23T16:00:00.000Z",  
          "updated_at": "2025-10-23T16:05:00.000Z",  
          "total_view_count": 10  
        }  
      ]  
    }  
  }  
}
```

```

{
  "id": 101,
  "album_id": 12,
  "title": "Come Together",
  "duration_seconds": 259,
  "order": 1,
  "label": ["Rock", "b", "c"],
  "view_count": 4,
  "is_cover": false,
  "lyrics": "Here come old flat top...",
  "cover_image_url": "/api/songs/101/cover",
  "created_at": "2025-10-20T10:01:00.000Z",
  "updated_at": "2025-10-20T10:01:00.000Z"
},
{
  "id": 100,
  "album_id": 10,
  "title": "Together",
  "duration_seconds": 248,
  "order": 1,
  "label": ["Rock"],
  "view_count": 2,
  "is_cover": false,
  "lyrics": "slowly...",
  "cover_image_url": "/api/songs/100/cover",
  "created_at": "2025-10-20T10:01:00.000Z",
  "updated_at": "2025-10-20T10:01:00.000Z"
}
],
...
]
}

```

12. 取得所有使用者

GET /api/users

Request

X-Authorization: Bearer <token>

Query Parameters:

- `cursor`: (string) e.g., "e2lkOiAxfQo="
- `limit`: (number) e.g., 10

Response (Success 200)

Content-Type: application/json

```
{
  "success": true,
  "data": [
    {
      "id": 1,
      "username": "user1",
      "email": "user1@web.wsa",
      "role": "user",
      "is_banned": false,
      "created_at": "2025-10-23T14:30:00.000Z"
    },
    {
      "id": 2,
      "username": "user2",
      "email": "user2@web.wsa",
      "role": "user",
      "is_banned": false,
      "created_at": "2025-10-23T15:00:00.000Z"
    },
    ... // the rest of 8 items
  ],
  "meta": {
    "next_cursor": "e2lkOiAxMX0K",
    "prev_cursor": "e2lkOiAxMH0=",
  }
}
```

13.更新使用者角色

系統必須至少保留一個管理員帳號。若最後一位管理員嘗試將自身角色變更為非管理員（例如：user）時，系統必須回傳錯誤。

不得更新被封鎖的使用者。

PUT /api/users/{user_id}

Request

X-Authorization: Bearer <token>

Content-Type: application/json

Route Parameters:

- `user_id`: (integer) e.g., 2

Body:

```
{  
  "role": "admin"  
}
```

Response (Success 200)

Content-Type: application/json

```
{  
  "success": true,  
  "data": {  
    "id": 2,  
    "username": "user2",  
    "email": "user2@web.wsa",  
    "role": "user",  
    "is_banned": false,  
    "created_at": "2025-10-23T15:00:00.000Z",  
    "updated_at": "2025-10-23T16:20:00.000Z"  
  }  
}
```

14. 封鎖使用者

不能封鎖自己

PUT /api/users/{user_id}/ban

Request

X-Authorization: Bearer <token>

Route Parameters:

- `user_id`: (integer) e.g., 2

Response (Success 200)

Content-Type: application/json

```
{  
  "success": true,  
  "data": {  
    "id": 2,
```

```
"username": "user2",
"email": "user2@web.wsa",
"role": "user",
"is_banned": true,
"updated_at": "2025-10-23T16:21:00.000Z"
}
```

15.解除封鎖使用者

PUT /api/users/{user_id}/unban

Request

X-Authorization: Bearer <token>

Route Parameters:

- `user_id`: (integer) e.g., 2

Response (Success 200)

Content-Type: application/json

```
{
  "success": true,
  "data": {
    "id": 2,
    "username": "user2",
    "email": "user2@web.wsa",
    "role": "user",
    "is_banned": false,
    "updated_at": "2025-10-23T16:22:00.000Z"
  }
}
```

16.創建新專輯

POST /api/albums

Request

X-Authorization: Bearer <token>

Content-Type: multipart/form-data

Body:

title	string
artist	string
release_year	number
genre	string
description	string

Response (Success 201 Created)

Content-Type: application/json

```
{
  "success": true,
  "data": {
    "id": 13,
    "title": "My New Album",
    "artist": "My Band",
    "release_year": 2025,
    "genre": "Indie Rock",
    "description": "This is the description of my new album.",
    "publisher": {
      "id": 2,
      "username": "user2",
      "email": "user2@web.wsa"
    },
    "created_at": "2025-10-23T16:00:00.000Z",
    "updated_at": "2025-10-23T16:00:00.000Z"
  }
}
```

17.更新專輯訊息

PUT /api/albums/{album_id}

Request

X-Authorization: Bearer <token>

Content-Type: application/json

Route Parameters:

- album_id: (integer) e.g., 13

Body:

title	string
description	string

Response (Success 200)

Content-Type: application/json

```
{
  "success": true,
  "data": {
    "id": 13,
    "title": "My Updated Album Title",
    "artist": "My Band",
    "release_year": 2025,
    "genre": "Indie Rock",
    "description": "Updated description.",
    "publisher": {
      "id": 2,
      "username": "user2",
      "email": "user2@web.wsa"
    },
    "created_at": "2025-10-23T16:00:00.000Z",
    "updated_at": "2025-10-23T16:05:00.000Z"
  }
}
```

18.刪除專輯

此刪除為軟刪除。

DELETE /api/albums/{album_id}

Request

X-Authorization: Bearer <token> Route Parameters: album_id: (integer) e.g., 13
Response (Success 200) Content-Type: application/json <pre>{ "success": true, }</pre>

19.新增歌曲到專輯

標籤必須以逗號分隔，且必須屬於預設標籤內的項目。

POST /api/albums/{album_id}/songs	
Request X-Authorization: Bearer <token> Content-Type: multipart/form-data Route Parameters: album_id: (integer) e.g., 13	
title	string
duration_seconds	number
label	string, optional
lyrics	string
cover_image	file, image/jpeg
is_cover	boolean
Response (Success 201 Created) Content-Type: application/json	

```

{
  "success": true,
  "data": {
    "id": 301,
    "album_id": 13,
    "title": "My First Song",
    "duration_seconds": 180,
    "lyrics": "Lyrics for the first song...",
    "order": 1,
    "view_count": 0,
    "label": ["a", "b", "c"],
    "is_cover": false,
    "cover_image_url": "/api/songs/301/cover",
    "created_at": "2025-10-23T16:10:00.000Z",
    "updated_at": "2025-10-23T16:10:00.000Z"
  }
}

```

20. 更新歌曲順序

更新歌曲順序時需要輸入專輯中的所有歌曲 id，並按照 body 輸入的順序調整專輯歌曲的排列順序。如果輸入的歌曲 id 與指定專輯歌曲不同、缺少歌曲，需要回傳錯誤訊息。

PUT /api/albums/{album_id}/songs/order

Request

X-Authorization: Bearer <token>

Content-Type: application/json

Route Parameters:

- album_id: (integer) e.g., 13

Body:

```

{
  "song_ids": [302, 301]
}

```

Response (Success 200)

Content-Type: application/json

```

{
  "success": true
}

```

21.更新歌曲訊息

標籤必須以逗號分隔，且必須為預設標籤清單中的項目。

POST /api/albums/{album_id}/songs/{song_id}	
Request	
X-Authorization: Bearer <token>	
Content-Type: multipart/form-data	
Route Parameters:	
- album_id: (integer) e.g., 13 - song_id: (integer) e.g., 301	
Body:	
title	string
duration_seconds	number
label	string
lyrics	string
cover_image	file, image/jpeg
is_cover	boolean
Response (Success 200)	
Content-Type: application/json	
<pre>{ "success": true, "data": { "id": 301, "album_id": 13, "title": "My First Song (Remix)", "duration_seconds": 190, "lyrics": "Lyrics for the first song...", "order": 2, "view_count": 0, "label": ["a","b","c"], "is_cover": false, "cover_image_url": "/api/songs/301/cover", "created_at": "2025-10-23T16:10:00.000Z", "updated_at": "2025-10-23T16:15:00.000Z" } }</pre>	

```
}
```

22. 自專輯刪除歌曲

此刪除為軟刪除。

DELETE /api/albums/{album_id}/songs/{song_id}
Request X-Authorization: Bearer <token> Route Parameters: • album_id: (integer) e.g., 13 • song_id: (integer) e.g., 301
Response (Success 200) Content-Type: application/json { "success": true, }

23. 取得專輯封面圖片連結

專輯封面連結將依照該專輯內歌曲的 is_cover 設定動態生成。

GET /api/albums/{album_id}/cover_url
Request Route Parameters: • album_id: (integer) e.g., 12
Response (Success 200) Content-Type: application/json {

```
    "success":true,
    "data":[
      "/api/songs/301/cover",
      "/api/songs/302/cover",
      "/api/songs/303/cover",
    ]
  }
```

24.取得歌曲封面圖片連結

GET /api/songs/{song_id}/cover_url

Request

Route Parameters:

- `song_id`: (integer) e.g., 101

Response (Success 200)

Content-Type: application/json

```
{
  "success":true,
  "data":{
    "cover_image_url": "/api/songs/301/cover",
  }
}
```

25.收藏歌曲

POST /api/songs/{song_id}/favorite

Request

X-Authorization: Bearer <token>

Route Parameters:

- `song_id`: (integer) e.g., 101

Response (Success 200)

Content-Type: application/json

```
{  
  "success": true,  
}
```

26. 取消收藏歌曲

POST /api/songs/{song_id}/unfavorite

Request

X-Authorization: Bearer <token>

Route Parameters:

- `song_id`: (integer) e.g., 101

Response (Success 200)

Content-Type: application/json

```
{  
  "success": true,  
}
```

27. 查看收藏清單

GET /api/favorites

Request

X-Authorization: Bearer <token>

Route Parameters:

Response (Success 200)

Content-Type: application/json

```
{
```

```
"success": true,
"data": [
  {
    "id": 102,
    "album_id": 12,
    "title": "Something",
    "label":["a","b","c"],
    "duration_seconds": 182,
    "album_title": "Abbey Road",
    "cover_image_url": "/api/songs/102/cover"
  },
  {
    "id": 205,
    "album_id": 11,
    "title": "Love of My Life",
    "label":["a","b","c"],
    "duration_seconds": 217,
    "album_title": "A Night at the Opera",
    "cover_image_url": "/api/songs/205/cover"
  }
],
}
```

28.新增並套用標籤

POST /api/tags	
Request	
X-Authorization: Bearer <token>	
Content-Type: multipart/form-data	
song_ids	array
label	string
Response (Success 200)	
Content-Type: application/json	
{ "success": true, "data": [{ "id": 102, "album_id": 12, "title": "Something",	

```
    "label":["a","b","c","new_tag"],
    "duration_seconds": 182,
    "album_title": "Abbey Road",
    "cover_image_url": "/api/songs/102/cover"
  },
  {
    "id": 205,
    "album_id": 11,
    "title": "Love of My Life",
    "label":["a","b","c","new_tag"],
    "duration_seconds": 217,
    "album_title": "A Night at the Opera",
    "cover_image_url": "/api/songs/205/cover"
  }
],
}
```

29. 移除標籤

DELETE /api/tags/{tag_id}
Request Route Parameters: tag_id: (integer) e.g., 101
Response (Success 200) Content-Type: application/json { "success": true, }

評分摘要

	子標準	分數
1	公開 API	8.5
2	使用者 API	5.25
3	管理員 API	10.25
總分		24.00