

Anime Basic API:

1. /api/anime/{id}

Description:

回傳對應ID的動畫資料

output format:

```
{ "Name": "One Piece",  
  "Score": 8.69,  
  "Category": ["Action", "Adventure", "Fantasy"],  
  "Description": "Gol D. Roger was known as the "Pirate King..."  
  "Type": "TV"  
  "Episodes": -1.0 // 不清楚、未結束用-1.0, 其他用一季的集數  
  "Air_Date": "1999/10/20"  
  "End_Date": "?" // 還未結束則用"?", 結束則跟Air Date同格式  
  "Image_URL": "https://...."  
}
```

2. /api/anime/all?sort={sort_type}

Description:

回傳所有種類的動畫ID, 按照sort_type排序, sort_type:
score_asc, score_desc, year_asc, year_desc

output format:

```
{ "ID": [1,2,4,8...] }
```

Categories API:

1. /api/category:

Description:

回傳所有的種類有哪些

output format:

```
{ "Categories": ["Love", "Fantasy"...] }
```

2. /api/category/{category}?sort={sort_type}

Description:

回傳特定種類的動畫ID, {category}內填入特定種類, 按照
sort_type排序, sort_type: score_asc, score_desc, year_asc,
year_desc

output format:

```
{ "ID": [1,2,4,8...] }
```

Search API:

1. /api/search/Hard/?keyword={keyword} (GET)

Description:

Hard Search, 回傳Name就是keyword的動畫ID
output format:
{"ID": {ID}} (沒東西的話, 會回傳空的dict)

2. /api/search/Soft/?keyword={keyword}&sort={sort_type} (GET)

Description:

Soft Search, 回傳Name或Description中有keyword的動畫ID, 按照sort_type排序, sort_type: score_asc, score_desc, year_asc, year_desc

output format:

{"ID": [1, 3, 8...]
"Name": ["abcd", "asfaha"]}

Auth API:

使用時 .env 檔裡面要加上 JWT_SECRET={隨便你打}

1. /api/auth/login (POST)

request body =

{"username": "{username}", "password": "{password}"}

output =

失敗: { error: 'Invalid username or password' }

成功: { message: 'Login successful', token }

#login token時長為30分鐘

2. /api/auth/search/{username} (GET)

Description:

回傳username = {username} 的 user資料

Output

{"id": "{id}", "username": "{username}", "email": "{email}"}

3. /api/auth/register (POST)

request body =

{"username": "{username}", "email": "{email}", "password": "{password}"}

output =

1. { error: 'All fields are required' } //有少填的欄位

2. { error: 'Username or email already registered!' }

//status是40

//信箱或名稱重複

3. { message: 'User registered successfully',

user: {

id: newUser.id,

username: newUser.username,

email: newUser.email

}} //註冊成功

4. /api/auth/update-password (PATCH)

```

request body =
    {"oldPW": "{oldPW}", "newPW": "{newPW}", "userID": "{userID}"}
//PW 是 password
output =
    1. { error: 'Both old and new passwords are required' }
    2. { error: 'Old password is incorrect' }
    3. { message: 'Password updated successfully' }
5. /api/auth/update-email (PATCH)
request body =
    {"PW": "{PW}", "newEmail": "{newEmail}", "userID": "{userID}"}
//PW 是 password
output =
    1. { error: 'Both password and new email address are required' }
    2. { error: 'Password is incorrect' }
    3. { message: 'Email updated successfully' }

```

Favorite API:

1. /api/favorite/create (POST):

Description:
創建收藏清單 (插入一個id為-1的動畫)

request body =

```

{"user_id": {user_id}, "list_title": "{list_title}"}

```

output format:

```

{"msg": "success", "error": "{error}"} 沒有error的話, error會留空字
串

```
2. /api/favorite/insert (POST):

Description:
插入動畫到某一清單

request body =

```

{"user_id": {user_id}, "list_title": "{list_title}", "anime_id":
{anime_id}}

```

output format:

```

{"msg": "success", "error": "{error}"}

```
3. /api/favorite/deleteList (POST):

Description:
刪除某一清單

request body =

```

{"user_id": {user_id}, "list_title": "{list_title}"}

```

output format:

```
{“msg”: “success”, “error”: “{error}”}
```

4. /api/favorite/deleteAnime (POST):

Description:

刪除清單中的某一動畫

request body =

```
{“user_id”: {user_id}, “list_title”: “{list_title}”, “anime_id”:  
{anime_id} }
```

output format:

```
{“msg”: “success”, “error”: “{error}”}
```

5. /api/favorite/getList (POST):

Description:

查看清單中有哪些動畫

request body =

```
{“user_id”: {user_id}, “list_title”: “{list_title}”}
```

output format:

```
{“anime_id”: [23, 424, 576], “anime_name”: [“Ring ni Kakero 1”,  
“Dirty Pair”, “Magikano”]}
```

6. /api/favorite/getUsersList (POST):

Description:

查看使用者有哪些清單, 回傳list_title

request body =

```
{“user_id”: {user_id}}
```

output format:

```
{“list_titles”: [“test”, “bbb”, “aaa”]}
```

Comment API:

1. /api/comment/add (POST):

Description:

對於某部動畫, 以某個使用者, 留下留言

request body =

```
{“user_id”: {user_id}, “anime_id”: {anime_id}, “content”:  
“{content}”}
```

output format:

```
{“msg”: “success”, “error”: “{error}”}
```

2. /api/comment/get (POST):

Description:

取得某部動畫底下的所有留言

request body =

```
{“anime_id”: {anime_id}}
```

output format:

```
{“user_id”: [1,3], “username”: [‘Xinil’, ‘Aokaado’], “content”: [“love  
this anime”, “Excellent anime”], comment_id: [2,3]}
```

3. /api/comment/deleteByID (POST):

Description:

用comment_id刪除某個留言

request body =

```
{“comment_id”: {comment_id}}
```

output format:

```
{“msg”: “success”, “error”: “{error}”}
```

4. /api/comment/deleteByInfo (POST):

Description:

用其他識別資訊刪除某個留言

request body =

```
{“user_id”: {user_id}, “anime_id”: {anime_id}, “content”:  
“{content}”}
```

output format:

```
{“msg”: “success”, “error”: “{error}”}
```

Score API:

1. /api/score/add (POST)

request body =

```
{“user_id”: {user_id}, “anime_id”: {anime_id}, “score”: “{score}”}
```

// score : 0 ~ 10

output format:

```
{“msg”: “success”, “error”: “{error}”}
```

2. /api/score/remove (POST)

request body =

```
{“user_id”: {user_id}, “anime_id”: {anime_id}}
```

output format:

```
{“msg”: “success”, “error”: “{error}”}
```

3. /api/score/getScore (POST)

get user's score on an anime

request body =

```
{“userID”: {userID}, “animeID”: {animeID}}
```

output format:

```
{“Score”: “{Score}”}
```