API Document

0.入口 2020-01-07

[Request Method] http://104.199.184.191:8080/edward

Request

HTTP Header {

token : “hfjdshfkjsdhf” (除了登入和註冊外所有API的請求 都要透過

TOKEN 來驗證, TOKEN放入的位置在 Header 裡面)

}

URL Parameters

Body (Type : Json)

Response

HTTP Status

Body (Type : Json)

1. 登入 2020-01-20

*\*備註 個人照片的 URL 前端可以去判斷要不要讀取 有需要讀取的狀況再去處理*

[POST] /login

Request

{

“email” : “test@example.com”,

“password” :　”123456”

}

Response

200

{

“token” : “ddsaasdsadad”,

“memberId” : “24324152134”,

“imageUrl” : “http://”

}

404

{

“message” : “Email unregistered or wrong password”

}

2.確認信箱是否已被註冊 2020-01-06

*\*備註 : 使用 Query Params 把需要確認的對象 直接放在URL*

[GET] /check/email?email=@example.comtest

Request

{

}

Response

200

{

“message” : “Email can be use”

}

409

{

message” : “Email is already registered”

}

3.註冊 2020-01-06

[POST] /register

Request

{

"email" : "test@example.com",

"password" : "123456",

"name" : "test123",

"birthDay" : "1990-01-01",

"gender" : 0,

"babies" : [

{"name" : "b1", "birthDay" : "20-20-20", "gender" : 0},

{"name" : "b2", "birthDay" : "20-20-20", "gender" : 1},

{"name" : "b3", "birthDay" : "20-20-20", "gender" : 0}

]

}

Response

204

{

}

409

{

message” : “Email is already registered”

}

4.取得會員基本資料 2020-01-07

*\*備註 : 透過 Path Variables 存放想要請求的對象*

*ex:/customers/33245/orders/8769/lineitems/1*

[GET] /members/{member\_id}/info

Request

{

}

Response

200

{

"name" : "test123",

"birthDay" : "1990-01-01",

"gender" : 0,

"babies" : [

{"name" : "b1", "birthDay" : "20-20-20", "gender" : 0},

{"name" : "b2", "birthDay" : "20-20-20", "gender" : 1},

{"name" : "b3", "birthDay" : "20-20-20", "gender" : 0}

]

}

401

{

message” : “Wrong validation”

}

404

{

message” : “Not found data”

}

5.修改會員資料 2020-01-07

[PUT] /members/{member\_id}/info

Request

{

"name" : "test123",

"birthDay" : "1990-01-01",

"gender" : 0,

"babies" : [

{"name" : "b1", "birthDay" : "20-20-20", "gender" : 0},

{"name" : "b2", "birthDay" : "20-20-20", "gender" : 1},

{"name" : "b3", "birthDay" : "20-20-20", "gender" : 0} ,

{"name" : "b2", "birthDay" : "20-20-20", "gender" : 1}

]

}

Response

204

{

}

401

{

message” : “Wrong vaildation”

}

404

{

message” : “Not found data”

}

6.取得隨機字卡 2020-01-23

*\*備註 : 使用 Query Params 把需要確認的對象 直接放在URL*

[GET] /cards?card\_pieces=2 \*備註:前端可以自己決定要回傳幾張

Request

{

}

Response

200

{

"cards" : [

{

“cardId” : “01”,

“title” : “卡片標題”,

“content” : “內文",

“issueTags” : [{“1”},{“2”}],

“ageTag” : “4-6",

“reference” : “XXX”,

“url” : “http/”,

“cardType” : 1

},

{

“cardId” : “02”,

“title” : “卡片標題”,

“content” : “內文",

“issueTags” : [{“1”},{“2”}],

“ageTag” “4-6",

“reference” : “XXX”,

“url” : “http/”,

“cardType” : 1

}

]

}

401

{

message” : “Wrong vaildation”

}

404

{

message” : “Not found data”

}

7.取得特定分類字卡 2020-01-20

*\*備註 : 使用 Query Params 把需要確認的對象 直接放在URL*

*\*備註 : a.用底線(“-”)表達月的區間*

\**備註: 如果只要請求一個Tag的話 就把另一個值改成 null*

*\*備註 : 要注意 "中文字" 在Http的URL裡面的格式*

[GET] /cards/tags?issueTags=健康&ageTag=3-5

Request

{

}

Response

200

{

"cards" : [

{

“cardId” : “01”,

“title” : “卡片標題”,

“content” : “內文",

“issueTags” : [{“1”},{“2”}],

“ageTag” : “4-6",

“reference” : “XXX”,

“url” : “http/”,

“cardType” : 1

},

{

“cardId” : “02”,

“title” : “卡片標題”,

“content” : “內文",

“issueTags” : [{“1”},{“2”}],

“ageTag” “4-6",

“reference” : “XXX”,

“url” : “http/”,

“cardType” : 1

}

]

}

401

{

message” : “Wrong vaildation”

}

404

{

message” : “Not found data”

}

8.收藏字卡 2020-01-11

*\*備註 使用者收藏字卡 或是 要上傳心得筆記 都使用此API*

[PUT] /members/{member\_id}/cards/{card\_id}

Request

{

"note” : “this is note”

}

Response

204

{

}

401

{

message” : “Wrong vaildation”

}

404

{

message” : “Not found data”

}

9.取消收藏 2020-01-11

[DELETE] /members/{member\_id}/cards/{card\_id}

Request

{

}

Response

204

{

}

401

{

message” : “Wrong vaildation”

}

404

{

message” : “Not found data”

}

10.創建自己的字卡 2020-01-26

*\*備註 以討論改成 上傳圖片的功能*

*\*備註 使用 multipart/form-data 的 Body 格式上傳圖片*

[POST] /members/{member\_id}/mycards

Header

{

*token : 43fdfdsfaf*

*Content-Type : multipart/form-data*

}

Request

{

*“file” : xxx.png (File 格式),*

*“note” : rtrewr (Text 格式)*

}

Response

204

{

}

401

{

message” : “Wrong vaildation”

}

404

{

message” : “Not found data”

}

11.新增自己創建字卡的Note 2020-01-21

*\*備註 新增或修改都是這個API*

[PUT] /members/{member\_id}/mycards/{mycard\_id}

Request

{

"note” : “this is note”

}

Response

204

{

}

401

{

message” : “Wrong vaildation”

}

404

{

message” : “Not found data”

}

12.刪除創建的字卡 2020-01-21

[DELETE] /members/{member\_id}/mycards/{mycard\_id}

Request

{

}

Response

204

{

}

401

{

message” : “Wrong vaildation”

}

404

{

message” : “Not found data”

}

13.請求使用者的所有收藏 2020-01-21

*\*備註 cardType: 1=知識卡片 2=自創卡片*

*\*備註 傳下來的自己創的卡片會給一個 檔案的 URL 透過那個連結去拿圖片*

[GET] /members/{member\_id}/cards

Request

{

}

Response

200

{

"cards" : [

{

“cardId” : “01”,

“title” : “卡片標題”,

“content” : “內文",

“issueTags” : [{“1”},{“2”}],

“ageTag” : “4-6",

“reference” : “XXX”,

“url” : “http/”,

“note” : “this is note”,

“time” : “2020-01-14”,

“cardType” : 1

},

{

“cardId” : “02”,

“title” : “卡片標題”,

“content” : “內文",

“issueTags” : [{“1”},{“2”}],

“ageTag” “4-6",

“reference” : “XXX”,

“url” : “http/”,

“note” : “this is note”,

“time” : “2020-01-14”,

“cardType” : 1

},

{

“cardId” : “111”,

“url” : “http/”,

“note” : “this is note”,

“time” : “2020-01-14”,

“cardType” : 2

},

{

“cardId” : “123”,

“url” : “http/”,

“note” : “this is note”,

“time” : “2020-01-14”,

“cardType” : 2

}

]

}

401

{

message” : “Wrong vaildation”

}

404

{

message” : “Not found data”

}

14.上傳個人大頭照 2020-01-23

[POST] /members/{member\_id}/image

Header

{

*token : 43fdfdsfaf*

*Content-Type : multipart/form-data*

}

Request

{

*“file” : xxx.png*

}

Response

200

{

“imageUrl” : “http//”

}

401

{

message” : “Wrong vaildation”

}

404

{

message” : “Not found data”

}

14.下載個人大頭照 2020-01-20

[GET] /members/{member\_id}/image

Request

{

}

Response

200

{

“imageUrl” : “http//”

}

401

{

message” : “Wrong vaildation”

}

404

{

message” : “Not found data”

}