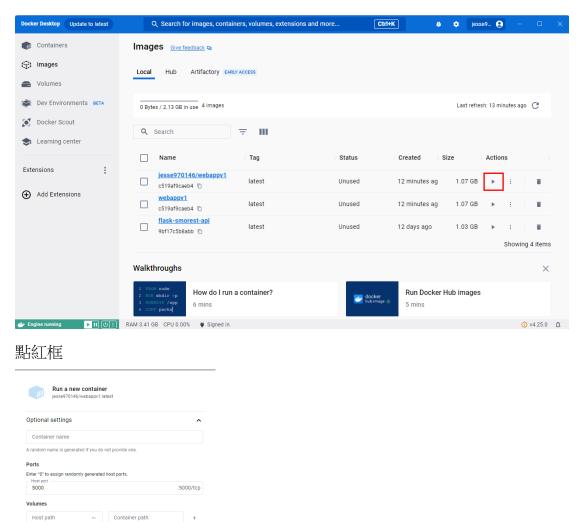
### Docker image: jesse970146/webappv1

### Docekr imgae 載下來後



Host port 打 5000

Value

Environment variables

Variable

api 使用的網址就會是 http://127.0.0.1:5000

Cancel

創帳號:/register

```
POST ▼ url /register

JSON ▼ Auth ▼ Query Headers 2 Docs

1 ↑ {
2 "username": "jesse",
3 "password": "testing"
4 }.
```

需要提供 username  $\operatorname{\mathbf{l}}$  password,username 是隨意,都可以,只要不重複

# 創建成功

```
{
   "message": "User created successfully."
}
```

# 重複名字

```
{
  "code": 409,
  "message": "A user with that username already exists.",
  "status": "Conflict"
}
```

### 登入: /login

也要提供 username 跟 password

成功登入會得到兩個 token:

### 登入失敗:

```
1 * {
2    "code": 401,
3    "message": "Invalid credentials",
4    "status": "Unauthorized"
5  }
```

登入後要保留 access\_token 作為 header 傳回來,可能向下圖這樣 Authorization 跟 Bearer 都是需要的

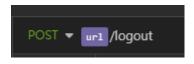
#### 1. 放在 HTTP Header 裡面

此方法為 Resource Server 資源服務器,為本系列文中所實作的方式,使用 GET 請求方式並將 API Token 放在 Header 裡。

```
GET /api/article/personal HTTP/1.1
Host: localhost:3000
Authorization: Bearer eyJhbGci0iJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJwYXlsb2FkIjp7InV
zZXJfaWQi0jEsInVzZXJfbmFtZSI6IkFuZHkxMCIsInVzZXJfbWFpbCI6ImFuZH1AZ21haWwuY29tI
n0sImV4cCI6MTUxNTcz0DUxOSwiaWF0IjoxNTE1NzM3NjE5fQ.CLPeXhcx12mdsL6-sUNFHFYABkTx
mzx3YxEPyNih_FM
```

然後 access\_token 跟 refresh\_token 都要保留,當 access\_token 過期時要拿 refresh\_token 跟我換新的 access\_toeken(/refresh),但這新的 access\_token 跟原本的比不夠 fresh,僅供一般普通瀏覽可以用,但如果要刪除一些東西,做破壞性的事情,就還是要請他登入,拿到一個 fresh 的 token

# 登出: /logout



只要提供 token 就好,因為 token 會帶 user id 的訊息,所以我可以知道哪個 user 登出

# 登出成功

```
1 * {
2     "message": " Sucessfully logout"
3 }
```

如果已經登出,token 會在 db 內紀錄,如果再次使用,會出現

```
1 * {
2    "descripition": "The token has been revoked",
3    "error": "token_revoked"
4  }
```

告訴使用者 token 已經不能使用

換 token:/refresh



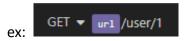
在 header 提供 refresh\_token,這邊會回傳一個不夠 fresh 的 token

### 沒提供 token

```
1 * {
2    "description": "Request does not contain an access token.",
3    "error": "authorization_required"
4  }
```

其他方便測試的,之後會刪掉

Get user: /user/id



可以得知創建的帳號

Delete user: /user/id

```
DELETE ▼ url /user/1
```

成功刪除

```
1 = {
2    "message": "User deleted."
3    }
```