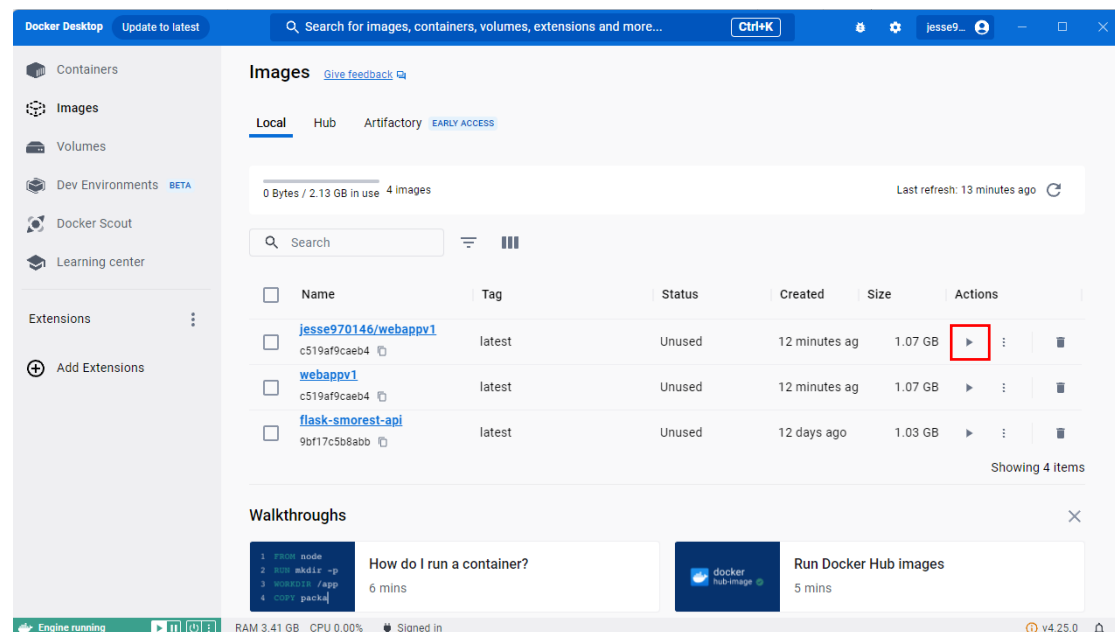
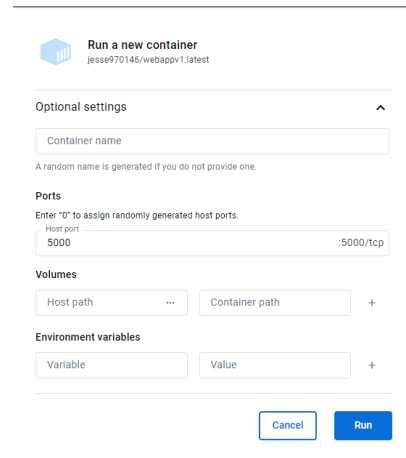


Docker image : jesse970146/webappv1

Docekr imgae 載下來後



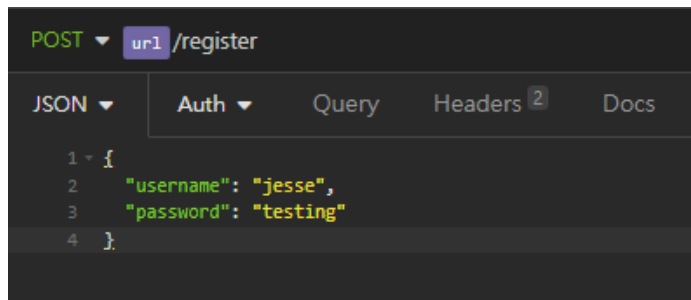
點紅框



Host port 打 5000

api 使用的網址就會是 <http://127.0.0.1:5000>

創帳號: /register



需要提供 username 跟 password，username 是隨意，都可以，只要不重複

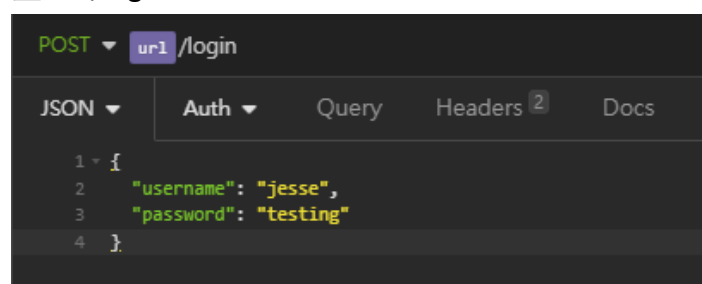
創建成功

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

重複名字

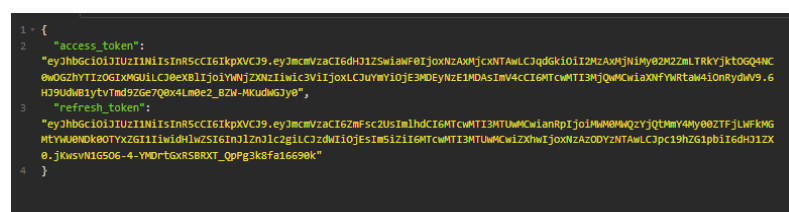
```
{
  "code": 409,
  "message": "A user with that username already exists.",
  "status": "Conflict"
}
```

登入: /login

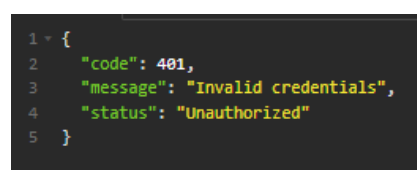


也要提供 username 跟 password

成功登入會得到兩個 token:



登入失敗:

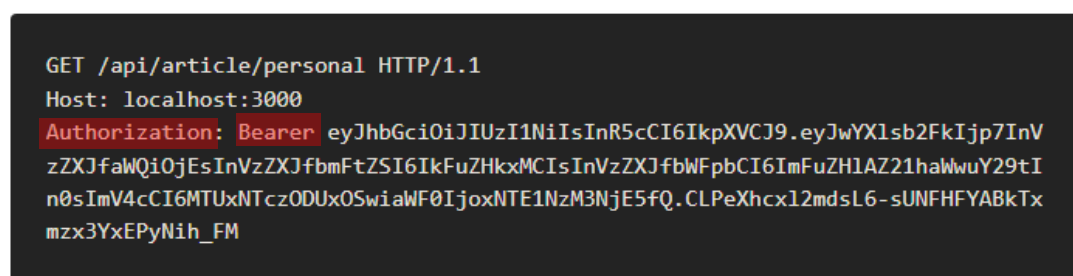


登入後要保留 `access_token` 作為 `header` 傳回來，可能向下圖這樣

Authorization 跟 Bearer 都是需要的

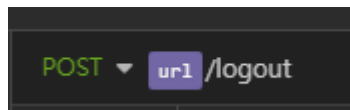
1. 放在 HTTP Header 裡面

此方法為 Resource Server 資源伺服器，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。



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_token(/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

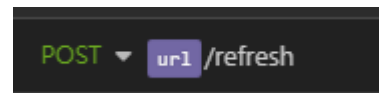
```
1 {  
2   "message": "Sucessfully logout"  
3 }
```

如果已經登出，token 會在 db 內紀錄，如果再次使用，會出現

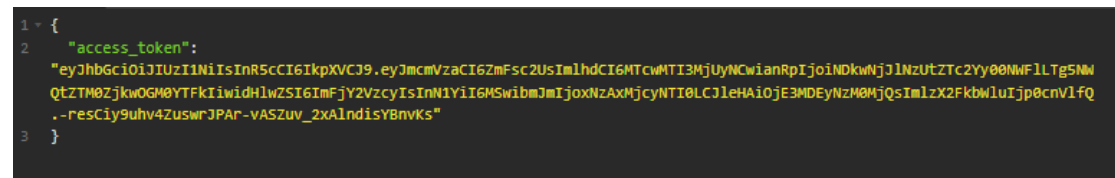
```
1 {  
2   "description": "The token has been revoked",  
3   "error": "token_revoked"  
4 }
```

告訴使用者 token 已經不能使用

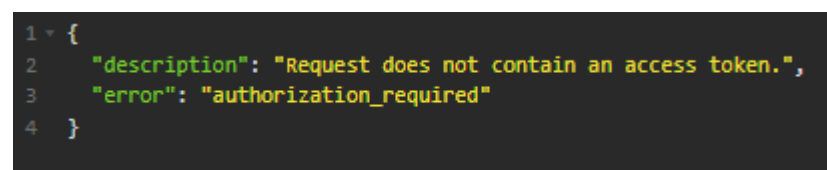
換 token : /refresh



在 header 提供 refresh_token，這邊會回傳一個不夠 fresh 的 token

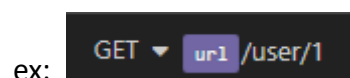


沒提供 token

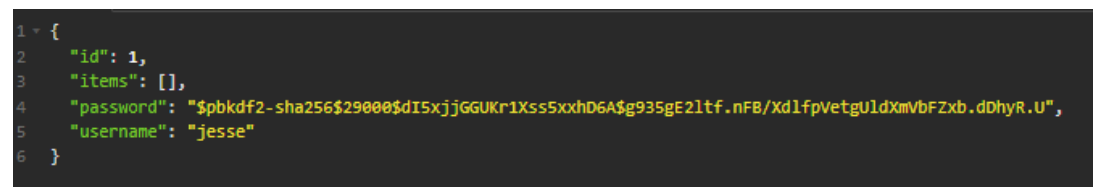


其他方便測試的，之後會刪掉

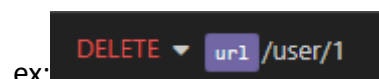
Get user: /user/id



可以得知創建的帳號



Delete user : /user/id



成功刪除

