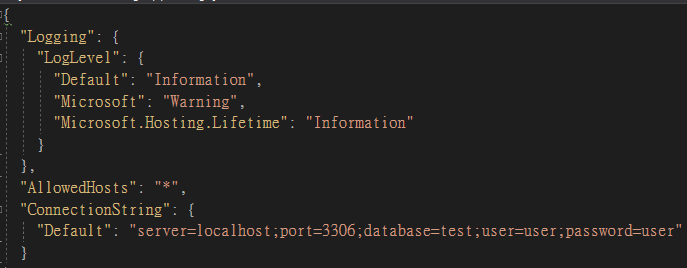
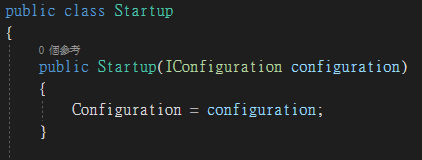
* 接出Configuration檔案

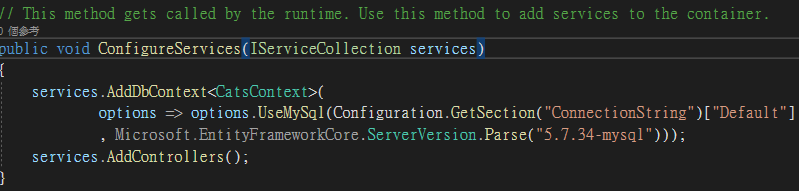
使用L3\_DBEntityFramework作為基底

於appsettings.json中加入Database相關參數

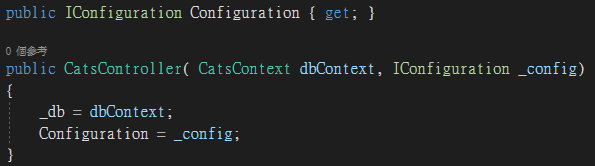


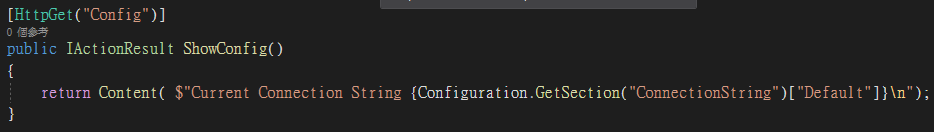
於Startup.cs->ConfigureService中的DbContext實體加入設定檔案



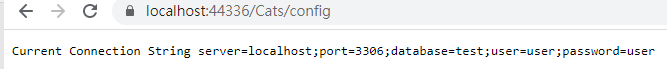


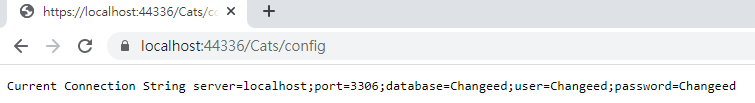
於Controller加入Configuration實體





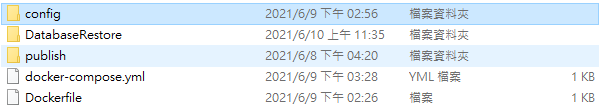
啟動網頁測試，可以看到設定檔改變讀出的文字也會改變



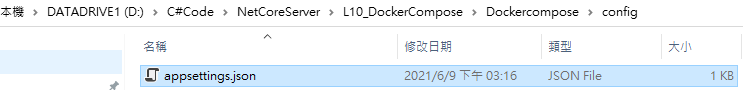


* 從Docker Container中接出Configuration檔案

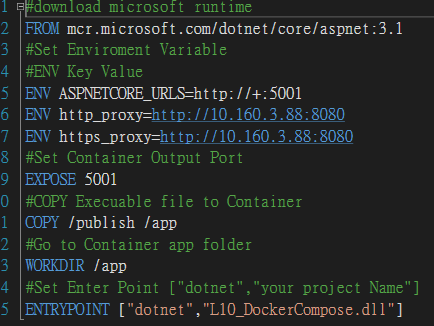
使用dotnet publish –c release –o publish輸出你的publish資料夾



創建一個config資料夾然後把publish中的appsettings.json複製進去



創建Dockerfile並用docker build -t server\_10 .產生images



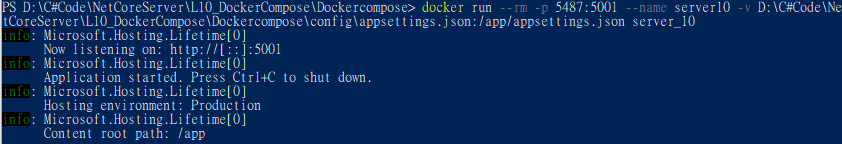




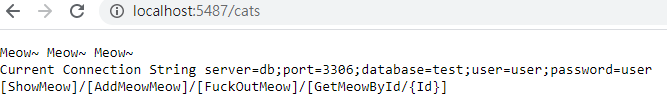
運行該container並使用-v將config檔案接出 (Config檔案為以host端為基準)

docker run –p [host port]:[container port] –v [host directory/file]:[container directory/file] [images]

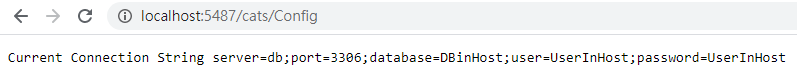
(docker run --rm -p 5487:5001 --name server10 -v D:\C#Code\NetCoreServer\L10\_DockerCompose\Dockercompose\config\appsettings.json:/app/appsettings.json server\_10 )



連入瀏覽器並測試網頁可不可以動



修改host端config中的appsetting.json後docker stop [Container]在重新docker start [Container]看會不會讀到 (我是看Config檔案有改但是不知道為什麼不會即時更新)



* 使用Docker Compose直接部屬整個系統

撰寫Docker-compose.yml檔案

定義services

**server**就是你的service Name

**image**就是用的images

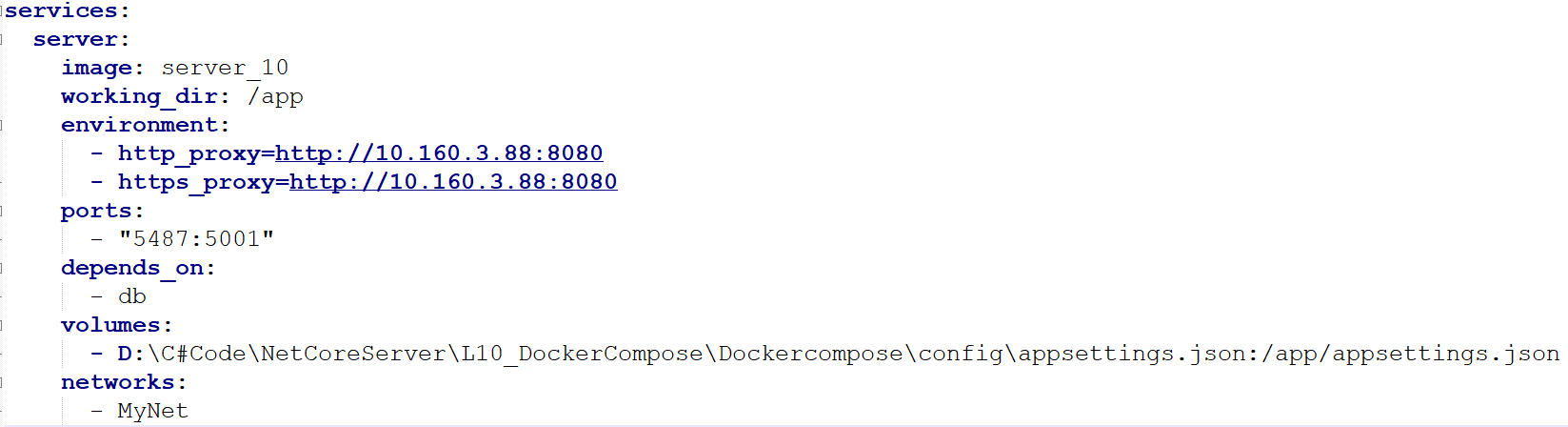
**environment**就是-e參數

**Port**就是-p參數

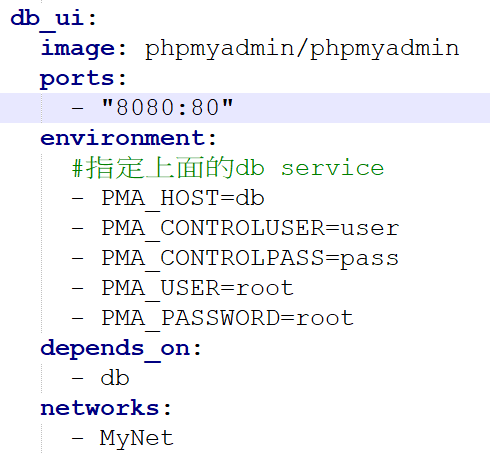
**depends**\_on就是他要等db service架起來才會架server這個service

**volumes**就是-v參數

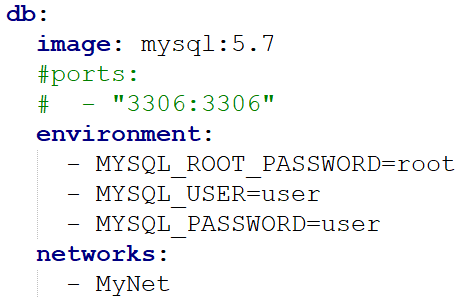
**networks**就是指定你要的網區，要同樣網區的才能用service name互相ping通



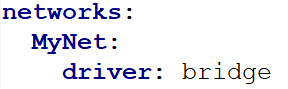
db\_ui Service



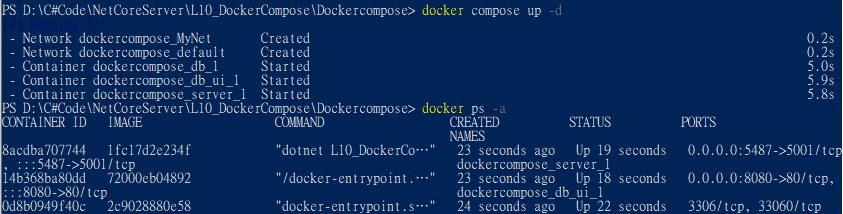
db Serivce



Network設定

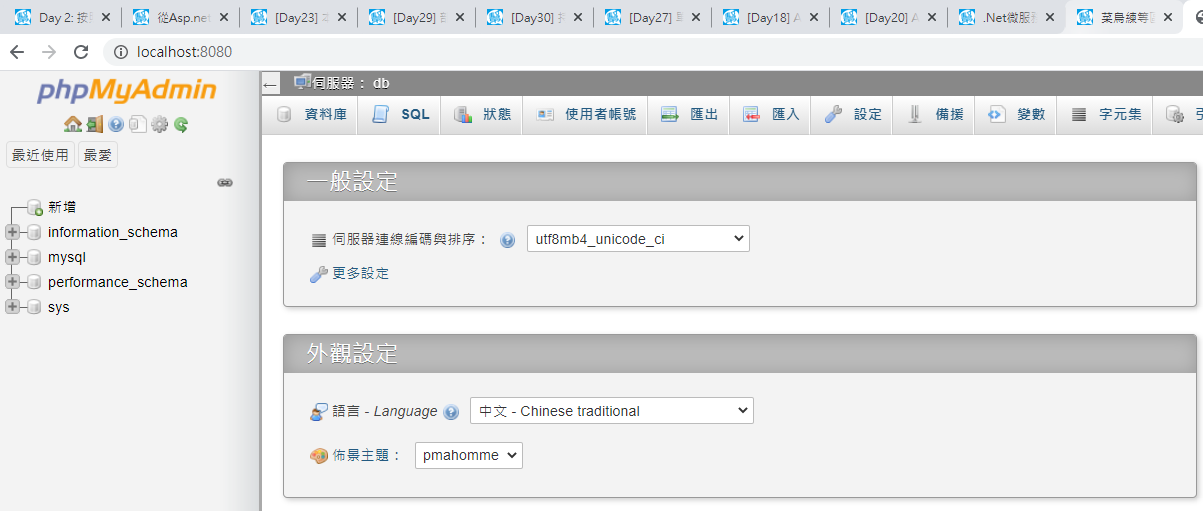


輸入docker compose up –d開始部屬

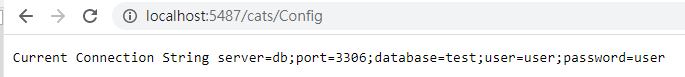


試著連一下網頁

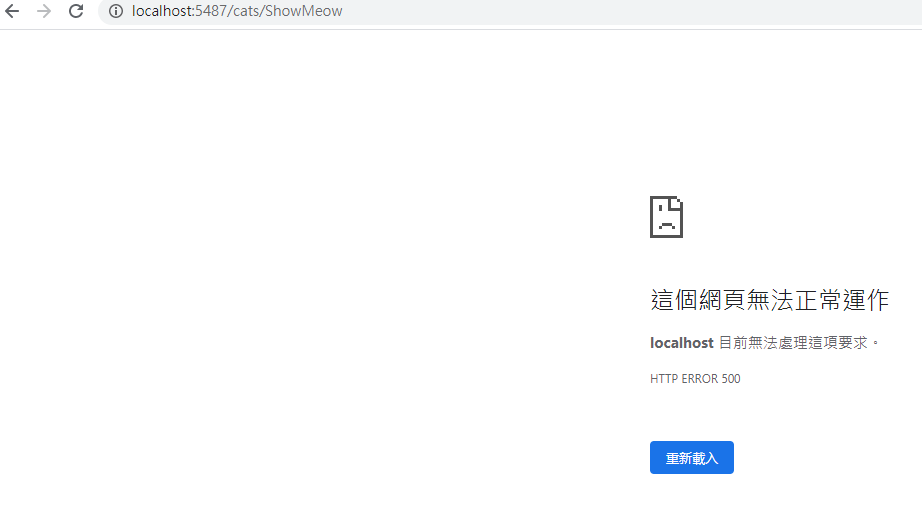
db\_ui



server

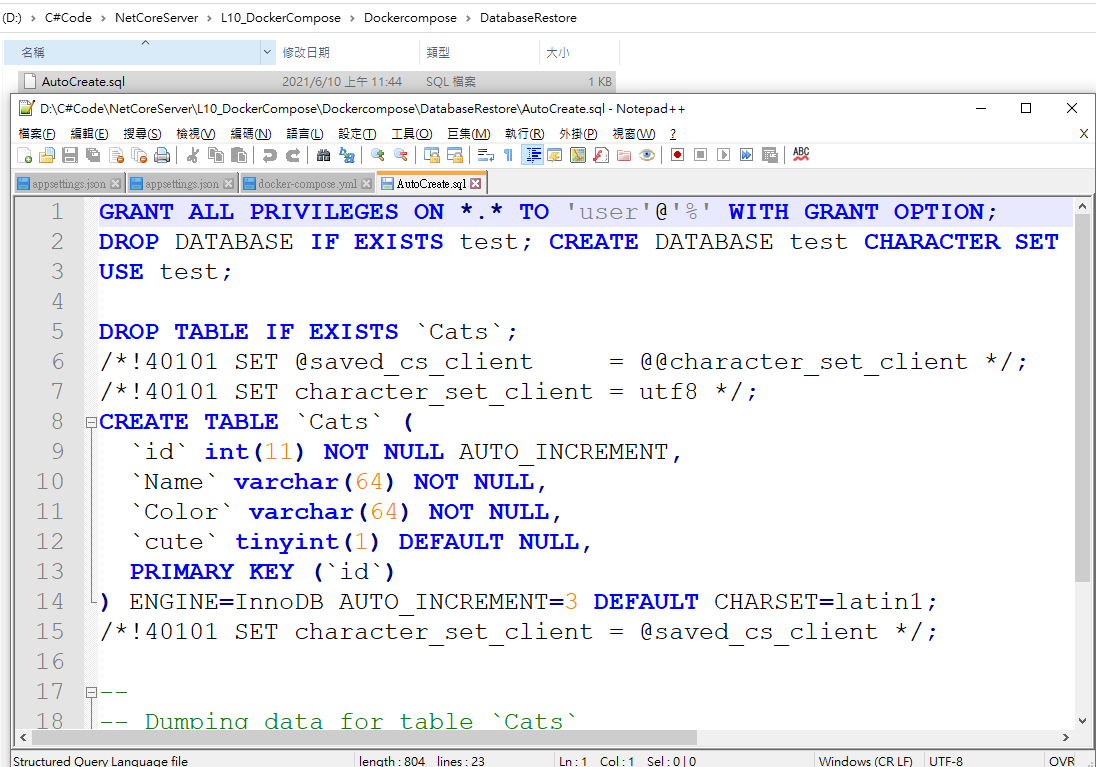


這時候你去連要取database資料的地方你會發現連不到，因為DB還沒建起來



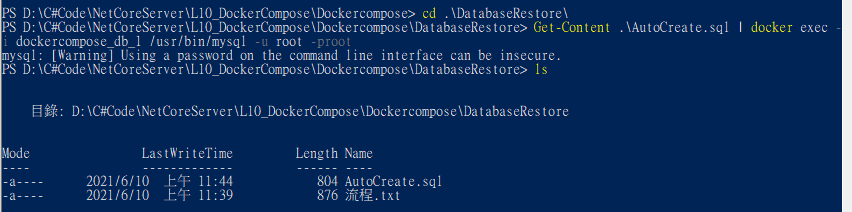
* restore復原你的資料庫資料

準備好你的sql檔案 (通常是使用mysqldump輸出來的，也可以自己寫)



使用pipline將你的資料匯入資料庫

Get-Content .\AutoCreate.sql | docker exec -i dockercompose\_db\_1 /usr/bin/mysql -u root –proot



檢查一下db\_ui看一下資料有沒有成功匯入



測試一下剛剛抓不到資料的頁面現在也可以讀取了

