3\_27

資料庫hash+salt可參考:

<https://sites.google.com/site/stevenattw/others/wang-zhan-mi-ma---hash-salt>

.net 3.0 安裝 :

1. 安裝dotnet-sdk-3.0.100-win-x64.exe
2. 在專案命令列key入:

dotnet add package Microsoft.EntityFrameworkCore.SqlServer --version 3.0.0

dotnet add package Microsoft.EntityFrameworkCore.Sqlite --version 3.0.0

dotnet tool install --global dotnet-ef --version 3.0.0

dotnet add package Microsoft.EntityFrameworkCore.Design --version 3.0.0

命令列:

1.在cmd下打入:

dotnet –info

node --version

npm –version

dotnet new –h

dotnet new webapi –h

dotnet new webapi -n –h

dotnet new webapi -n DatingApp.API

2.在VSCODE內

Ctrl+S 存檔

Ctrl+shift+P

Ctrl+P

Ctrl+` Terminal

dotnet run

dotnet watch run

打prop 按tab鍵,則會生出public int MyProperty { get; set; }

打 ctor 按tab鍵,則會生出 public ClassName(Parameters) 建構式

dotnet ef –h

dotnet ef migrations add InitialCreate

dotnet ef migrations remove

dotnet ef database update

angular

npm install -g @angular/cli

Run

git config --global user.email "you@example.com"

git config --global user.name "Your Name"

設定成:

git config --global user.email "c6312900@gmail.com"

git config --global user.name "Bill Fu"

github 帳號: [c6312900@gmail.com](mailto:c6312900@gmail.com) 密碼:Bill7321$

使用者名稱:c6312900

2-13 2:10 package.json 記載node\_modules的版本號

2-15 10:19 alt+O 在TYPESCRIPT中開啟檔案

2-16 1:05 在DatingApp專案中startup.cs 內加入services.AddCors();

2-20 2:11 npm install bootstrap font-awesome

2-21 Git

2-21 2:27 git status

2-21 5:27 git init

2-21 7:17 initial Commit

2-21 9:13 git remote add origin https://github.com/c6312900/DatingApp.git

3-25 2:12 dotnet ef migrations add AddedUserEntity

3-25 2:58 dotnet ef database update

3-32 3:05 Debug

1.先在Terminal頁籤跑程式dotnet run

2.先設中斷點,按VSCODE最中左邊有1隻蟲的按鈕,然後在左邊上方, 綠色箭頭旁,選擇Add Configuration🡪.NET Core🡪選擇.NET Core Attach, KEY入該程式DatingApp.API.exe

3.去postname看

3-35 00:44 若是 netcoreapp3.0 擇要安裝packages 按

Ctrl+shift+p選 NuGet Package Manager : Add Package

輸入Microsoft.identityModel 選 Microsoft.identityModel.Tokens

選最新版本(目前是5.6.0) ,接著再安裝第2個package ,

輸入System.identityModel.Tokens.Jwt, 選最新版本(目前是5.6.0)

3-36 00:44 安裝packages Microsoft.AspNetCore.Authentication.JwtBearer

3-37 1:10 在DattingApp.API找.gitignore 內加入appsettings.json

進入Terminal 進入 DattingApp.API

鍵入git rm appsettings.json 有錯誤 改鍵入git rm appsettings.json

(此段測試git有錯誤,後參考<https://noootown.wordpress.com/2015/06/19/git-first-use/> git push origin master 解決)

3-37 2:44 usersecrets 觀念是在開發模式,將資料庫連線字串或”AppSettings:Token” “Super secrect Key” 等密鑰放在secrets.json檔案中,

該檔案放在環境變數中,為何放在環境變數中因為by每個使用者環境設定,該檔案不放在專案中,放在每個使用者自己的環境變數中,因此git到雲端時,該檔案不會被傳上(程式碼不包含此檔案),所以確保了該檔案內容的安全

secrets.json檔案放的位置

Windows:%APPDATA%\microsoft\UserSecrets\<userSecretsId>\secrets.json

上述路徑有一段 <userSecretsId> 嗎？這段會是系統產生的 GUID 值，Secret Manager 會使用這組 userSecretsId 來將你的專案及 secret.json 綁在一起

專案檔 .csproj，裡面的 PropertyGroup 中會新增一段 UserSecretsId 節點，這裡面的值就是對應到上述路徑的userSecretsId

詳細可參考

<https://poychang.github.io/microsoft-user-secret-manager-with-dotnet-core/>

<https://www.itread01.com/content/1546145503.html>

<https://docs.microsoft.com/zh-tw/aspnet/core/security/app-secrets?view=aspnetcore-3.1&tabs=windows>

4-43 1:23 this.auth.login(this.model).subscribe(), 最後的括號 ( 要自己key入,則可看到overloads 覆載滑鼠點在向下的箭頭即可看到其他的覆載

也可將游標移到括號內按ctrl+shift+space鍵

4-43 5:12 查token網址 <https://jwt.io>

4-44 2:40 key入 div.dropdown-divider