1. Mkdir Reactivities
2. dotnet new sln
3. dotnet new webapi -n API
4. dotnet new classlib -n Application
5. dotnet new classlib -n Domain
6. dotnet new classlib -n Persistence
7. dotnet sln add API/API.csproj 或 dotnet sln add API
8. dotnet sln add Application
9. dotnet sln add Domain
10. dotnet sln add Persistence
11. dotnet sln list
12. cd API 設定reference

dotnet add reference ../Application

1. cd Application 設定reference

dotnet add reference ../Persistence

dotnet add reference ../Domain

1. cd persistence 設定reference

dotnet add reference ../Domain

1. cd api
2. dotnet run 或dotnet watch run
3. 在activity.cs內可用prop後按tab產生屬性
4. ctrl+shift+P 打入nuget 選

NuGet open nuget gallery,

輸入microsoft.entityframeworkcore.sqlite 選擇

並在右上方選在Persistence專案,5.0.1版本

1. 在Persistence專案->DataContext.cs->游標放在DbContext

再找黃色燈泡選產生建構函式(DataContext(options)

2-11 在API下打入clear,停止API SERVER , cd..

dotnet tool list --global 內dotnet-ef要5.0.0以上

去<https://www.nuget.org/packages/dotnet-ef> 找

如有舊版先解除dotnet tool uninstall --global dotnet-ef

安裝新版dotnet tool install --global dotnet-ef --version 5.0.4

或使用 dotnet tool update --global dotnet-ef --version 5.0.4

建立migration

dotnet ef migrations add InitialCreate -s API -p Persistence

ctrl+shift+P 打入nuget 選

NuGet open nuget gallery,

輸入microsoft.entityframeworkcore.design 選擇

並在右上方選在API專案,5.0.4版本 install

在執行 dotnet ef migrations add InitialCreate -s API -p Persistence

2-12 dotnet ef database -h

2-15(git相關看此篇) 在D:\Project\Reactivities>git init

dotnet new -l (忽略檔案)

2-18 node --version

npm --version

npx create-react-app client-app --use-npm --template typescript

cd client-app 接著 npm start

3-26 cd client-app 接著 npm install axios

3-27 cors policy 進入API找startup.cs檔案,在

ConfigureServices(IServiceCollection services)內

加入

services.AddCors(opt => {

                opt.AddPolicy("CorsPolicy", policy => {

                    policy.AllowAnyMethod().AllowAnyHeader().WithOrigins("http://localhost:3000");

                });

            });

在

public void Configure(IApplicationBuilder app, IWebHostEnvironment env)

內在app.UseRouting() 後加入 app.UseCors(“CorsPolicy”)

3-28 去<https://react.semantic-ui.com/usage/>

npm install semantic-ui-react semantic-ui-css

4-31 Clean Architecture 架構

Entity 🡪Domain 資料庫table的Entity 例如: public class Activity

Use Cases 🡪business logic 有許多的class例如:edit Entity,add Entity,

delete Entity

 we can do is get a list of our activities and we can get the details of an individual activity

controllers (interface adapters)🡪負責user interface 和use case溝通

user interface

4-32 架構 可看 1:44

4-33 APIControllers are going to get their data ,

Application project to handle These requests,

It is going to contain all of our business logic

在application 中安裝NUGet ctrl+shift+p 選Open NugGet Gallery輸入

Mediator 選 Mediatr.Extensions.Microsoft.DependencyInjection 在右邊勾選application.csproj

4-38在application 中安裝NUGet ctrl+shift+p 選Open NugGet Gallery輸入

AutoMap 選 **AutoMapper.Extensions.Microsoft.DependencyInje** 在右邊勾選application.csproj

若程式無法using Automapper 則在terminal 輸入 dotnet restore 再重開專案

4-42 Debugger 可看這集

1. 先在程式中選擇要中斷的點,就是紅色的點,再將Terminal進入API內,選擇最左邊那一欄,有1個三角形旁邊有蟲的按鈕按下,接著在左上方有一個綠色三角形選.NET Core Lanch(web) ,直接按下綠色三角形即可,接著用postman送出request則會在中斷點停
2. 另一種方式先在程式中選擇要中斷的點,就是紅色的點,再將Terminal進入API內執行dotnet watch run,接著選擇最左邊那一欄,有1個三角形旁邊有蟲的按鈕按下,接著在左上方有一個綠色三角形選.NET Core Attach ,直接按下綠色三角形,接著會有選擇框,選擇API.exe或API.dll,接著用postman送出request則會在中斷點停 ,另外如果要結束,可善右上方的行,找到紅色的鈕(Disconnect)按下去

5-57 npm install uuid

npm i --save-dev @types/uuid