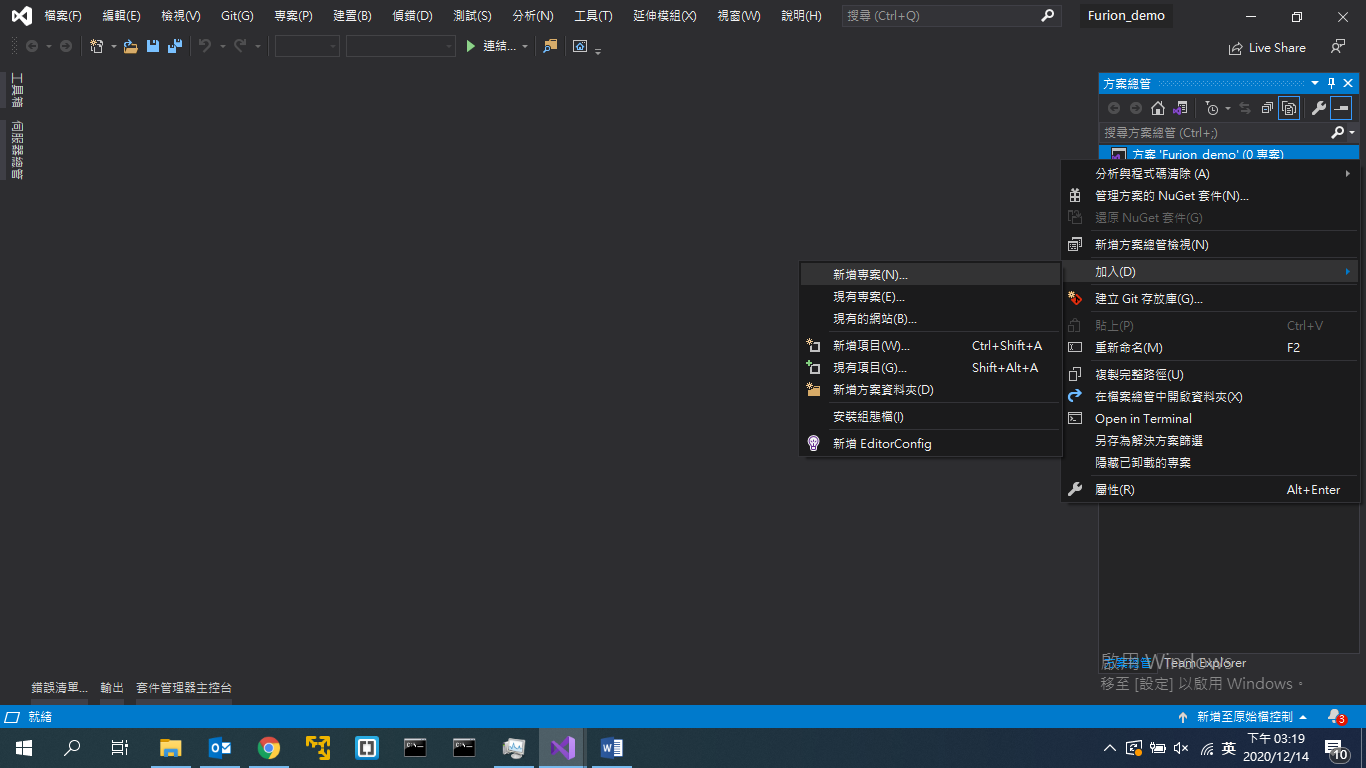
**架設Furion環境**

Visual Studio建立一空白方案

新增五個類別庫

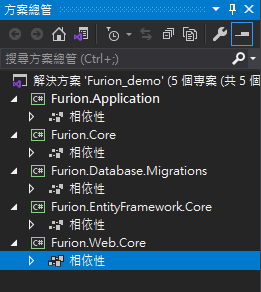
分別為:

* Furion.Application
* Furion.Core
* Furion.Database.Migrations
* Furion.EntityFramework.Core
* Furion.Web.Core





建立後刪除附加的.cs檔



再新增一專案 ASP.NET Core Web應用程式

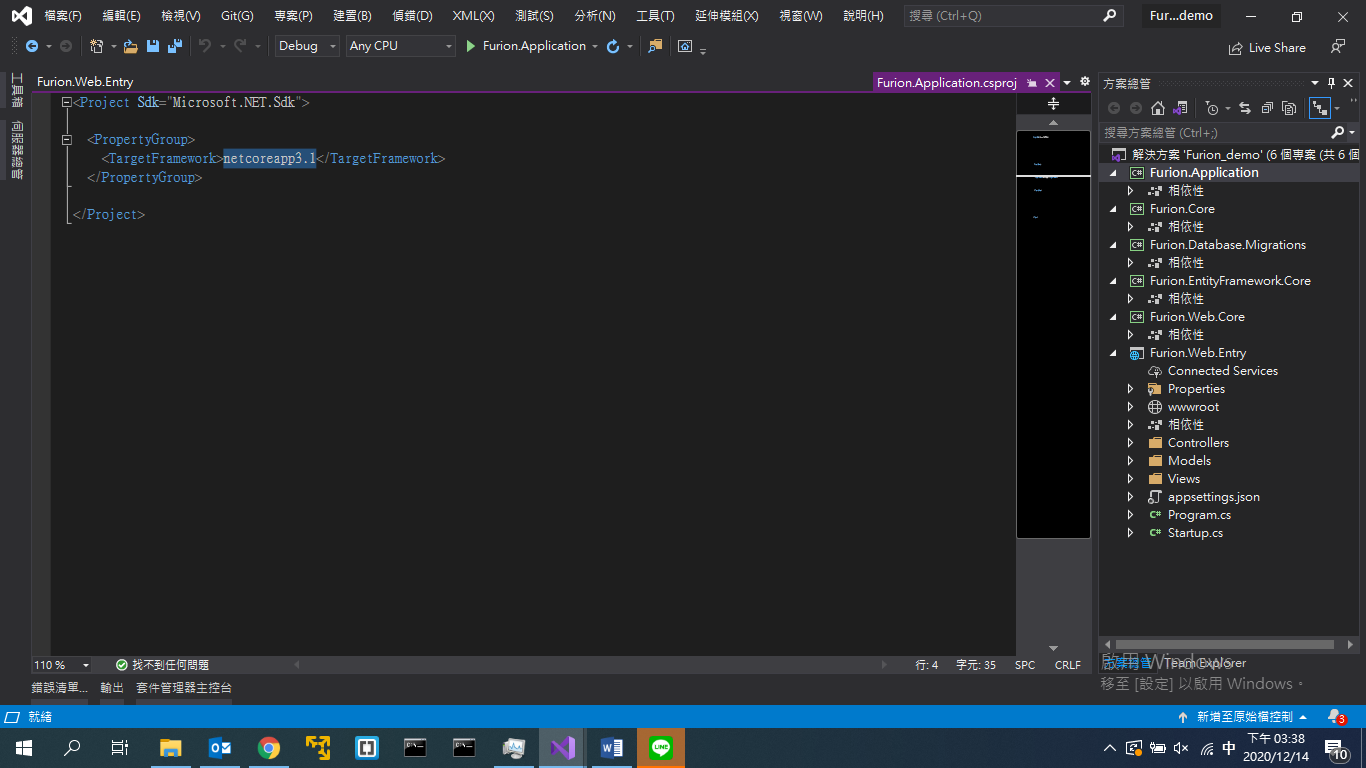
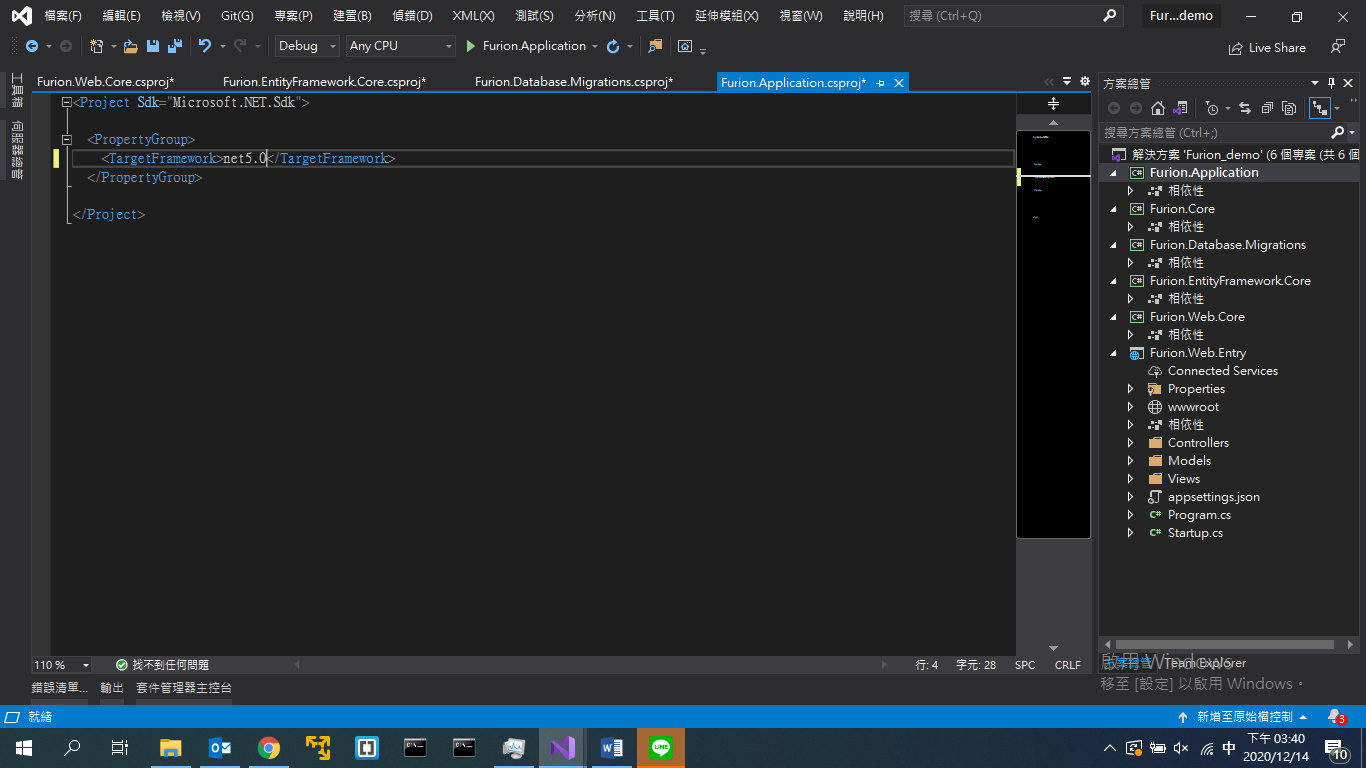


命名為Furion.Web.Entry

選擇MVC架構

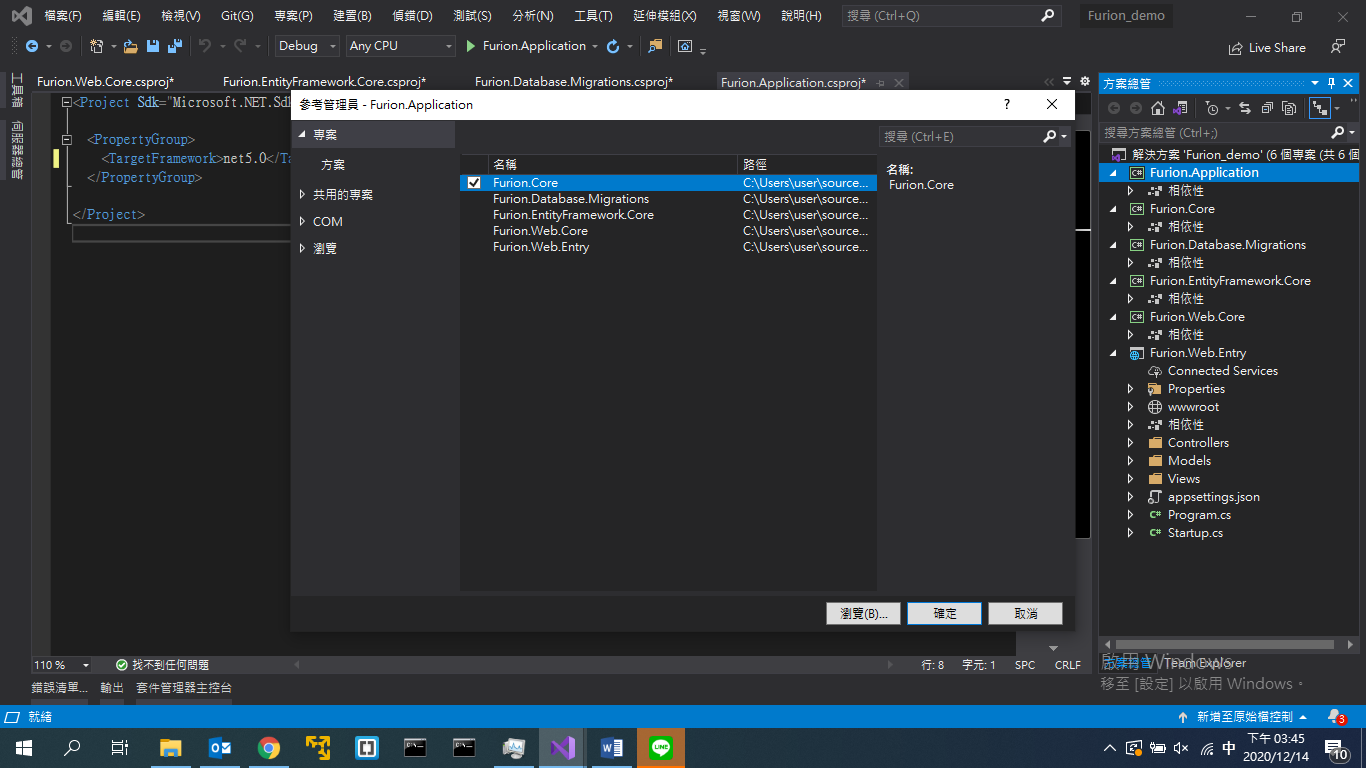
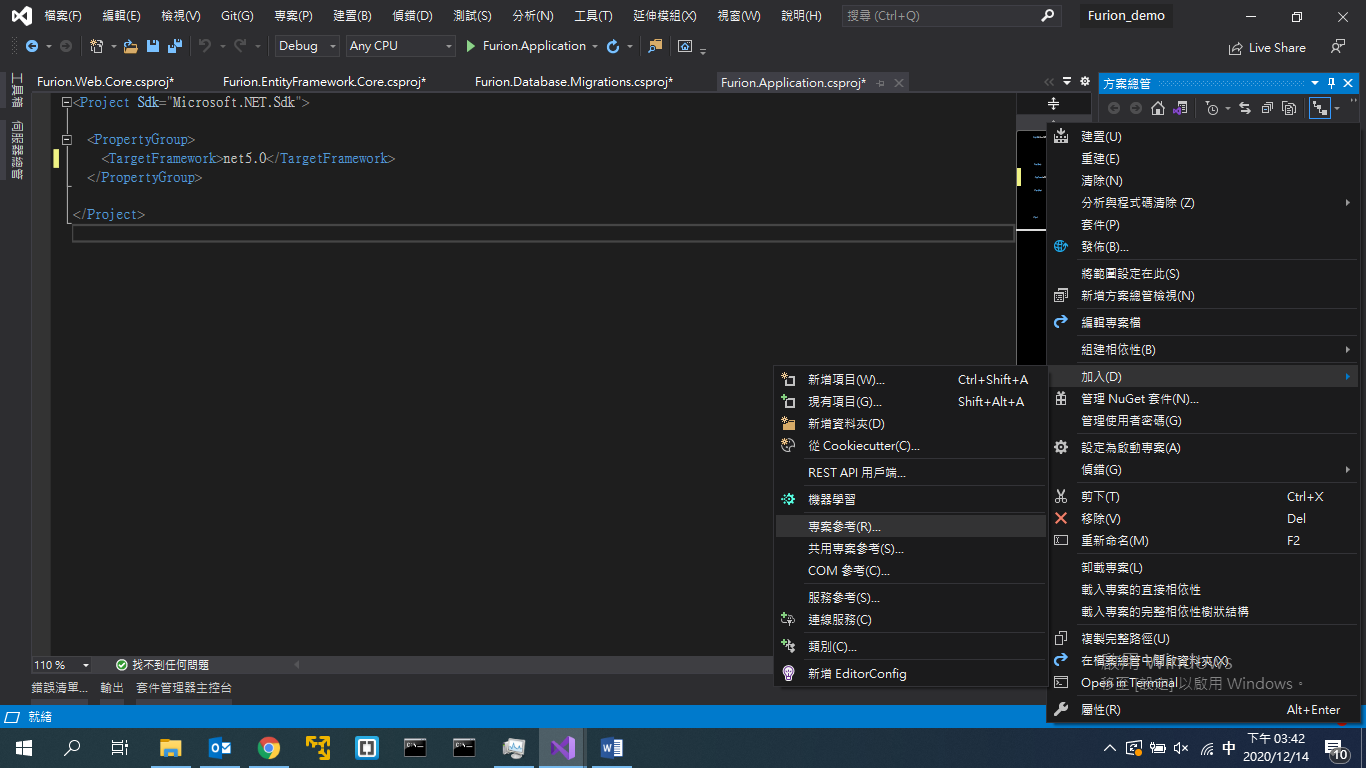


點擊各個專案並將版本改為net5.0

解決方案-->右鍵-->**重建方案**

右鍵Furion.Application加入專案參考，增加Furion.Core



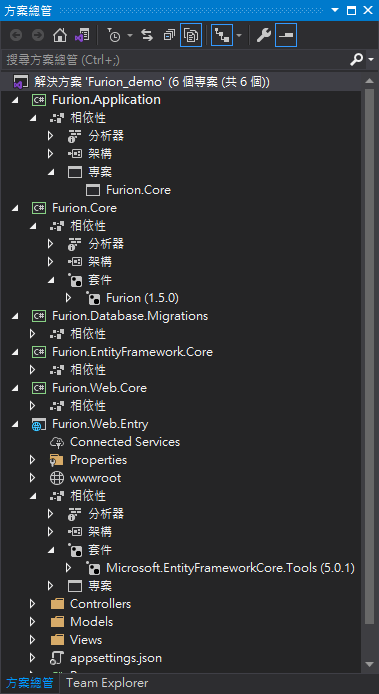
其餘專案參考如下

* Furion.Application：添加 Furion.Core 參考
* Furion.Database.Migrations：添加 Furion.EntityFramework.Core  參考
* Furion.EntityFramework.Core：添加 Furion.Core  參考
* Furion.Web.Core：添加 Furion.Application, Furion.Database.Migrations 參考
* Furion.Web.Entry：添加 Furion.Web.Core  參考

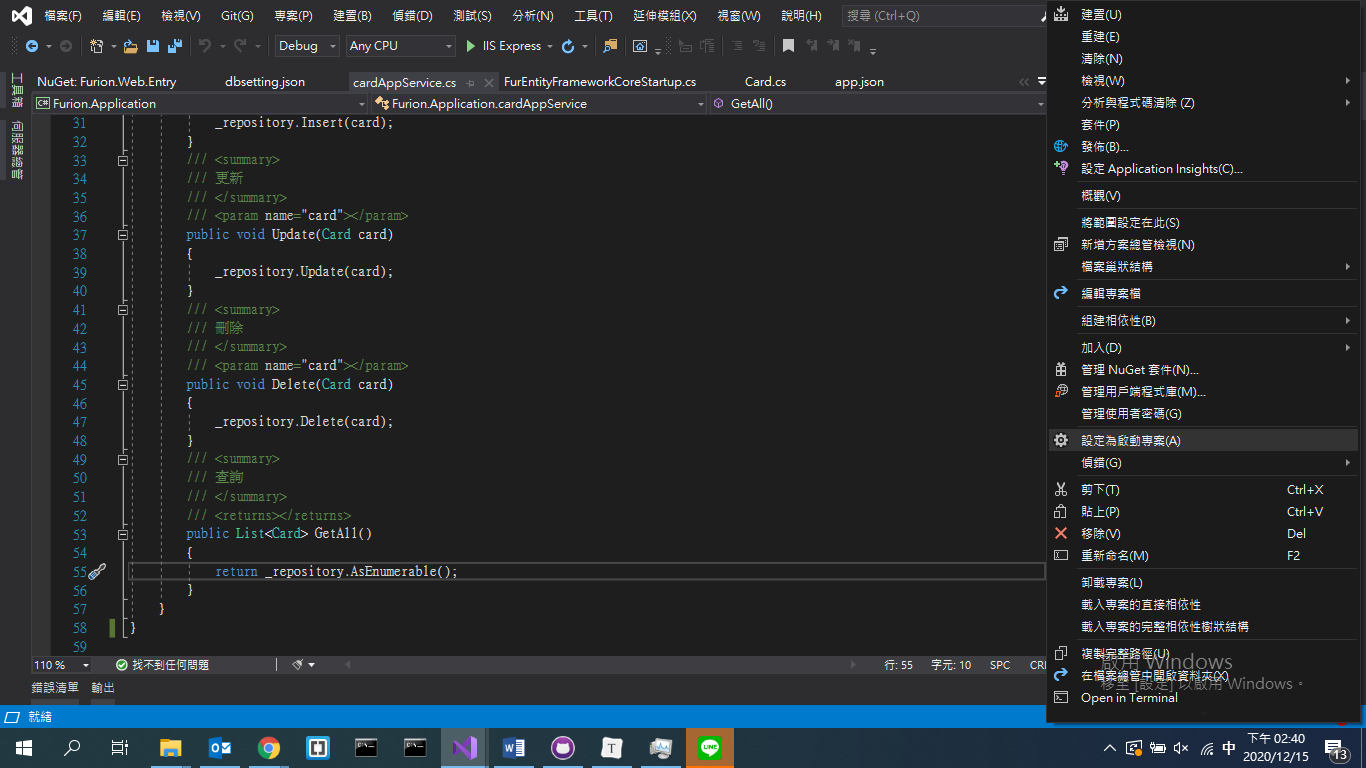
在Furion.Core🡪右鍵🡪管理NuGet套件🡪安裝Furion

在Furion.Web.Entry🡪右鍵🡪管理NuGet套件🡪安裝Microsoft.EntityFrameworkCore.Tools

**重建解決方案**



將Furion.Web.Entry設為啟動專案



於*Furion.Web.Entry/Program.cs*中添加Inject()

**public** static IHostBuilder CreateHostBuilder**(**string**[]** args**)** **=>**

Host**.**CreateDefaultBuilder**(**args**)**

**.**ConfigureWebHostDefaults**(**webBuilder **=>**

**{**

webBuilder**.**Inject**().**UseStartup**<**Startup**>();** **//here**

**});**

於*Furion.Web.Entry/Startup.cs*中

在ConfigureServices中添加AddInject()

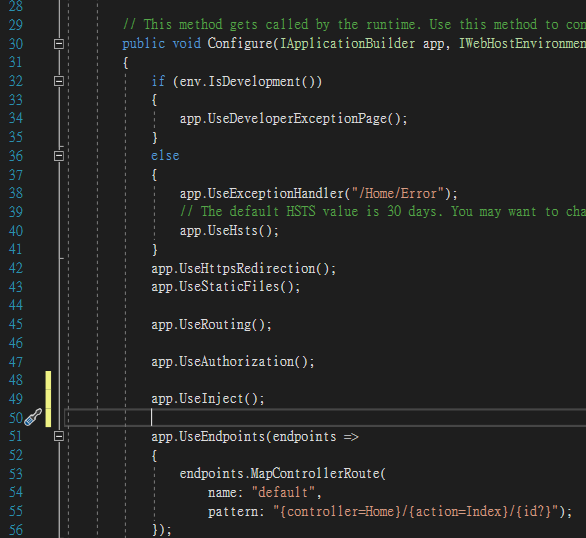
**public** void ConfigureServices**(**IServiceCollection services**)**

**{**

services**.**AddControllersWithViews**().**AddInject**();//here**

**}**

在Configure中添加app.UseInject()



於Furion.Web.Core新增一類別, 命名為FurionWebCoreStartup.cs

改變Startup.cs與FurionWebCoreStartup.cs的程式碼如下

*Furion.Web.Entry/Startp.cs*

**using** Microsoft**.**AspNetCore**.**Builder**;**

**using** Microsoft**.**AspNetCore**.**Hosting**;**

**using** Microsoft**.**Extensions**.**Configuration**;**

**using** Microsoft**.**Extensions**.**DependencyInjection**;**

**namespace** Furion**.**Web**.**Entry

**{**

**public** class Startup

**{**

**public** Startup**(**IConfiguration configuration**)**

**{**

Configuration **=** configuration**;**

**}**

**public** IConfiguration Configuration **{** **get;** **}**

**public** void ConfigureServices**(**IServiceCollection services**)**

**{**

**}**

**public** void Configure**(**IApplicationBuilder app**,** IWebHostEnvironment env**)**

**{**

**}**

**}**

**}**

*Furion.Web.Core/FurionWebCoreStartup.cs*

**using** Microsoft**.**AspNetCore**.**Builder**;**

**using** Microsoft**.**AspNetCore**.**Hosting**;**

**using** Microsoft**.**Extensions**.**DependencyInjection**;**

**using** Microsoft**.**Extensions**.**Hosting**;**

**namespace** Furion**.**Web**.**Core

**{**

**public** class FurionWebCoreStartup**:**AppStartup

**{**

**public** void ConfigureServices**(**IServiceCollection services**)**

**{**

services**.**AddControllersWithViews**().**AddInject**();**

**}**

**public** void Configure**(**IApplicationBuilder app**,** IWebHostEnvironment env**)**

**{**

**if** **(**env**.**IsDevelopment**())**

**{**

app**.**UseDeveloperExceptionPage**();**

**}**

**else**

**{**

app**.**UseExceptionHandler**(**"/Home/Error"**);**

app**.**UseHsts**();**

**}**

app**.**UseHttpsRedirection**();**

app**.**UseStaticFiles**();**

app**.**UseRouting**();**

app**.**UseAuthorization**();**

app**.**UseInject**();**

app**.**UseEndpoints**(**endpoints **=>**

**{**

endpoints**.**MapControllerRoute**(**

name**:** "default"**,**

pattern**:** "{controller=Home}/{action=Index}/{id?}"**);**

**});**

**}**

**}**

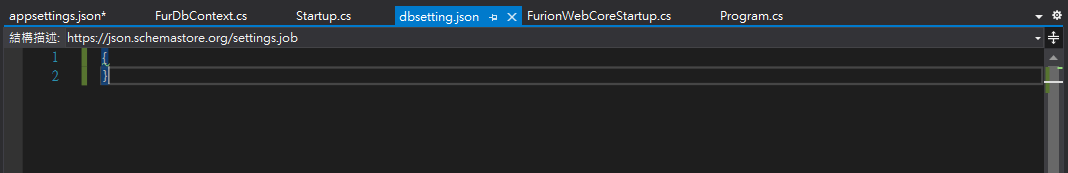
**}**

**連接資料庫**

**配置連接字串**

在Furion.EntityFramework.Core新建檔案FurDbcontext.cs和dbsetting.json

*dbsetting.json*結構描述設定為https://json.schemastore.org/appsetting



*Furion.EntityFramework.Core/dbsetting.json*

**{**

"ConnectionStrings"**:** **{**

"DbConnectionString"**:** ";Data Source=user;Initial Catalog=test;Integrated Security=True"

**}**

**}**

*Furion.EntityFramework.Core/FurDbcontext*

**using** Furion**.**DatabaseAccessor**;**

**using** Microsoft**.**EntityFrameworkCore**;**

**namespace** Furion**.**EntityFramework**.**Core

**{**

**[**AppDbContext**(**"DbConnectionString"**)]** *// 支持 appsetting.json名或連接字串*

**public** class FurDbContext **:** AppDbContext**<**FurDbContext**>**

**{**

**public** FurDbContext**(**DbContextOptions**<**FurDbContext**>** options**)** **:** **base(**options**)**

**{**

**}**

**}**

**}**

**重建方案**

**AppStartup配置**

Furion.EntityFramework.Core建新檔FurionEntityFrameworkCoreStartup.cs

*Furion.EntityFramework.Core/FurionEntityFrameworkCoreStartup*

**using** Furion**.**DatabaseAccessor**;**

**using** Microsoft**.**Extensions**.**DependencyInjection**;**

**namespace** Furion**.**EntityFramework**.**Core

**{**

**[**AppStartup**(**600**)]** *//數值越大則越先執行*

**public** **sealed** class FurEntityFrameworkCoreStartup **:** AppStartup

**{**

**public** void ConfigureServices**(**IServiceCollection services**)**

**{**

services**.**AddDatabaseAccessor**(**options **=>**

**{**

options**.**AddDbPool**<**FurDbContext**>(**DbProvider**.**SqlServer**);**

*//連甚麼資料庫就選甚麼*

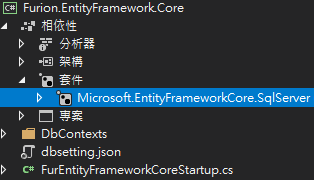
**},** migrationAssemblyName**:**"Furion.Database.Migration"**);**

**}**

**}**

**}**

在Furion.EntituFramework.Core安裝Microsoft.EntityFrameworkCore. SqlServer (依據連接的資料庫做更改)



**資料庫上下文定位器**

*Furion.Core/FurDbContextLocator.cs*

**using** Furion**.**DatabaseAccessor**;**

**namespace** Furion**.**Core

**{**

**public** **sealed** class FurDbContextLocator **:** IDbContextLocator

**{**

**}**

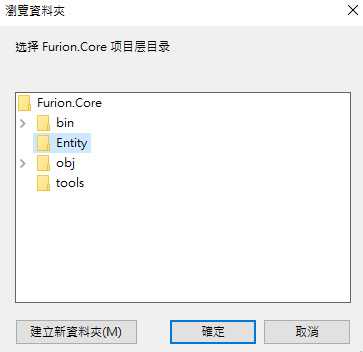
**}**

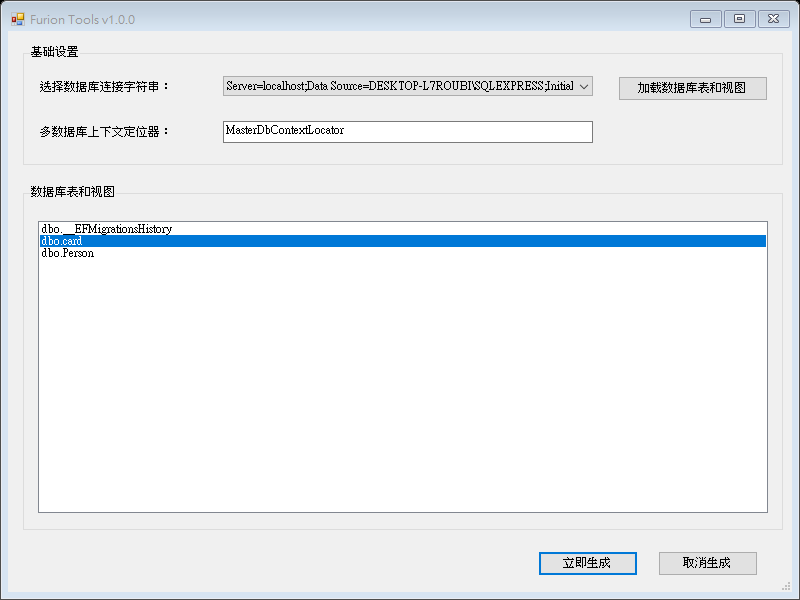
**重建方案**

**資料庫生成模型**

1. Furion.Core新增一資料夾Entity
2. 於Furion的Github下載tools資料夾並放在專案的Furion.Core的資料夾
3. 工具🡪NuGet套件管理員🡪套件管理器主控台🡪預設專案設定為Furion.Core
4. 輸入 &"C:\Users\user\source\repos\Furion\_demo\Furion.Core\tools\cli.ps1" (cli.ps1之完整路徑位置)
5. 輸入G🡪選擇連接字串🡪加載數據庫和視圖🡪選擇資料庫🡪立即生成🡪

選擇Entity資料夾

****



生成後腳本會自動在Entity資料夾內生成.cs檔

**重建方案**

**數據庫操作基本設定**

Furion.Application新增AppService.cs

*Furion.Application/AppService.cs*

**using** Furion**.**Core**;**

**using** Furion**.**DatabaseAccessor**;**

**using** Furion**.**DynamicApiController**;**

**using** System**.**Collections**.**Generic**;**

**namespace** Furion**.**Application

**{**

*/// <summary>*

*/// DB資料庫*

*/// </summary>*

**public** class cardAppService **:** IDynamicApiController

**{**

**private** **readonly** IRepository**<**Card**>** \_repository**;**

*/// <summary>*

*///*

*/// </summary>*

*/// <param name="repository"></param>*

**public** cardAppService**(**IRepository**<**Card**>** repository**)**

**{**

\_repository **=** repository**;**

**}**

*/// <summary>*

*/// 新增*

*/// </summary>*

*/// <param name="card"></param>*

**public** void insert**(**Card card**)**

**{**

\_repository**.**Insert**(**card**);**

**}**

*/// <summary>*

*/// 更新*

*/// </summary>*

*/// <param name="card"></param>*

**public** void Update**(**Card card**)**

**{**

\_repository**.**Update**(**card**);**

**}**

*/// <summary>*

*/// 刪除*

*/// </summary>*

*/// <param name="card"></param>*

**public** void Delete**(**Card card**)**

**{**

\_repository**.**Delete**(**card**);**

**}**

*/// <summary>*

*/// 查詢*

*/// </summary>*

*/// <returns></returns>*

**public** List**<**Card**>** GetAll**()**

**{**

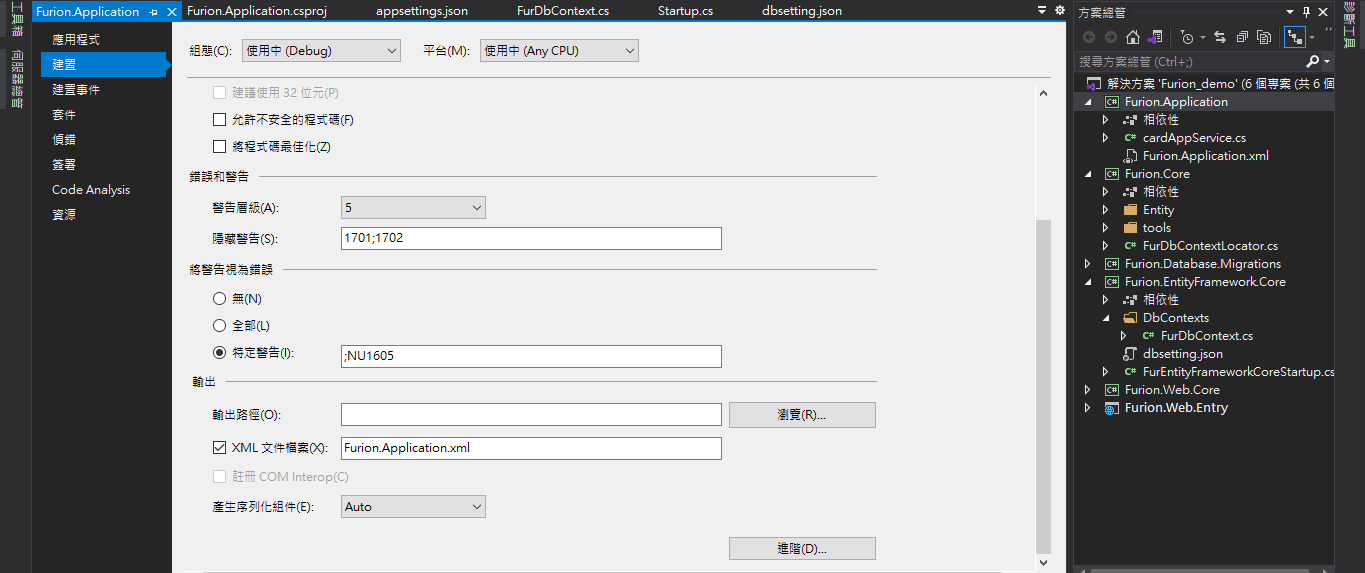
**return** \_repository**.**AsEnumerable**();**

**}**

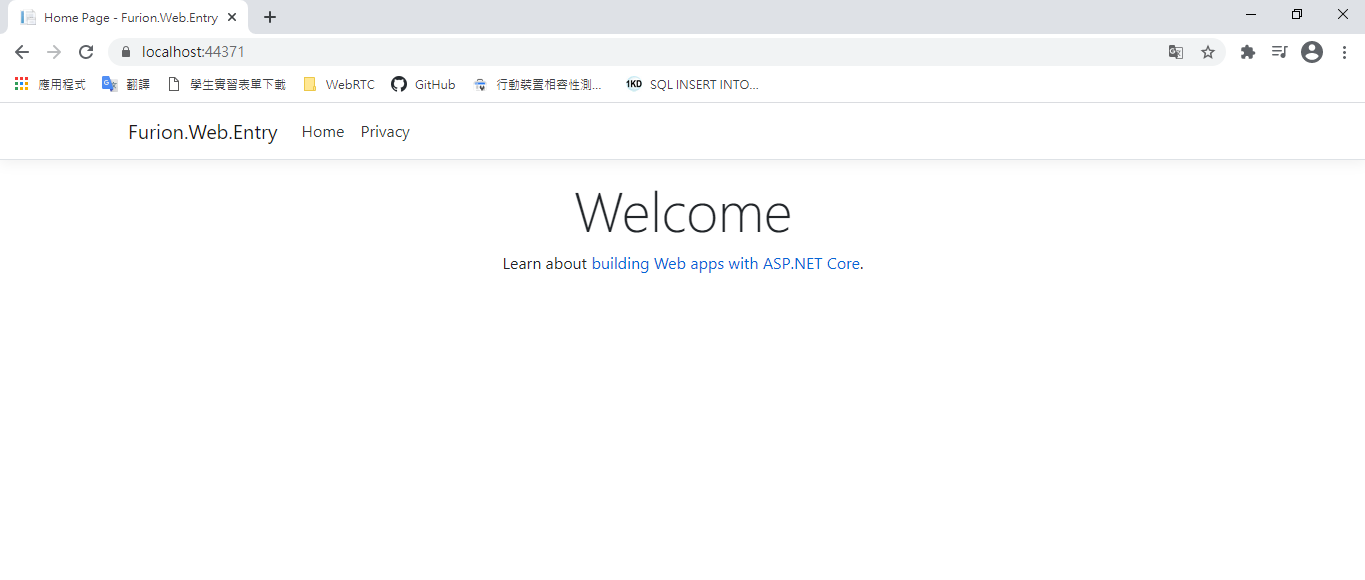
**}**

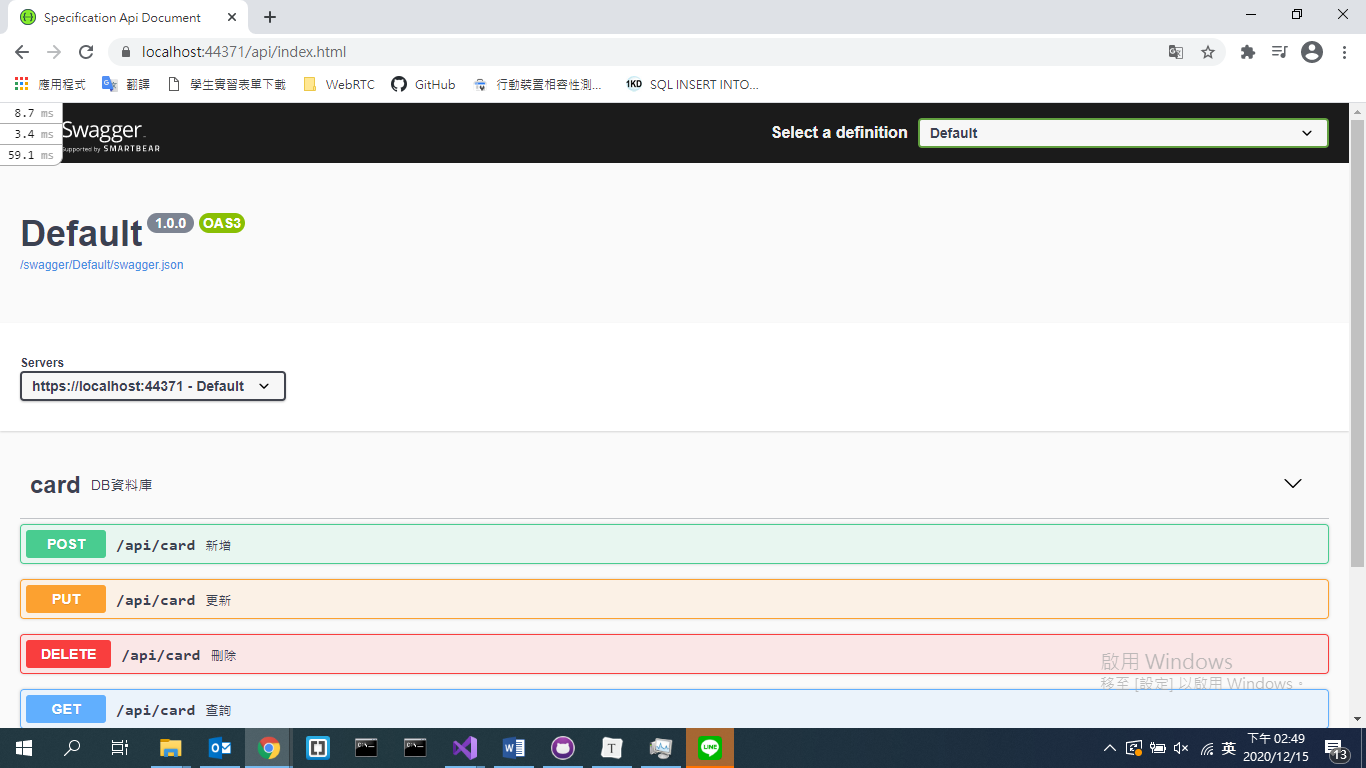
**}**

Furion.Application右鍵🡪屬性🡪建置🡪打勾XML文件檔案🡪名為Furion.Application.xml

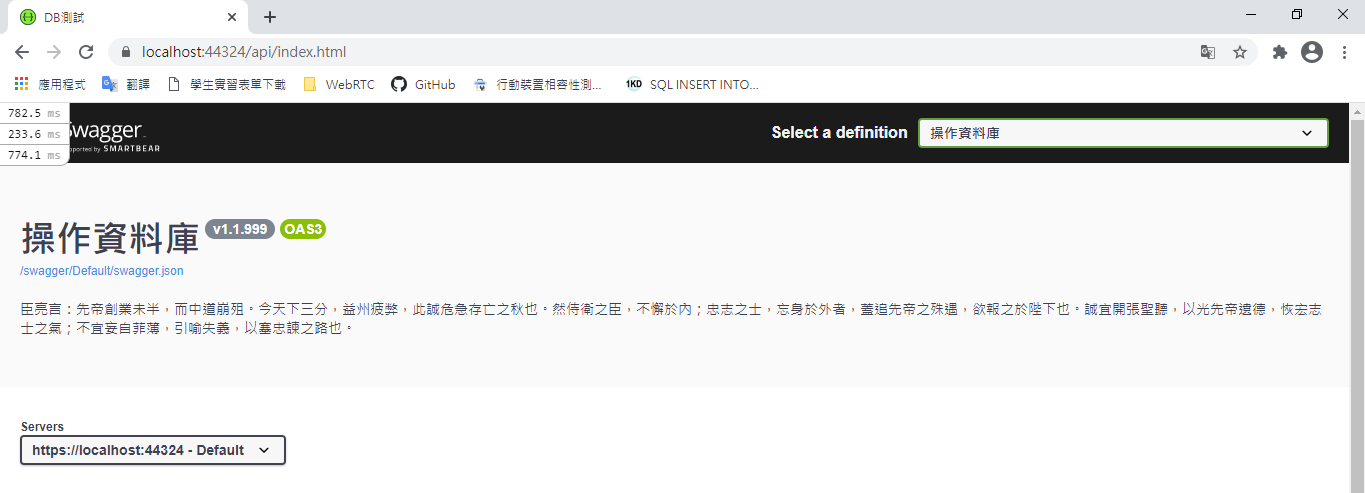
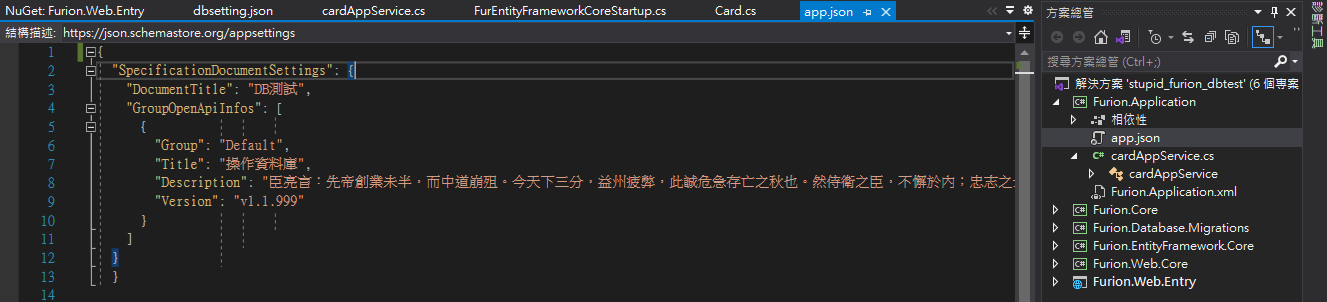


**重建方案**

**執行**

在網址後輸入 /api

**Swagger版面設計**

Furion.Application新增app.json