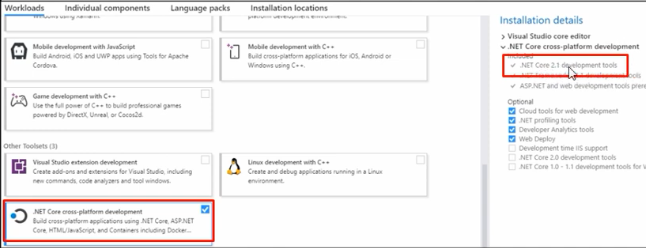
MB

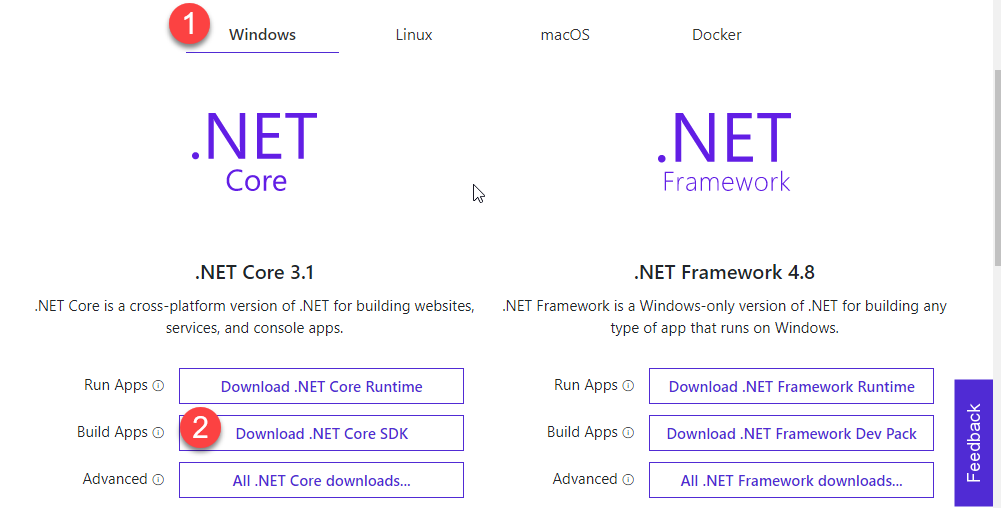
################################################################

ASP Core Web API

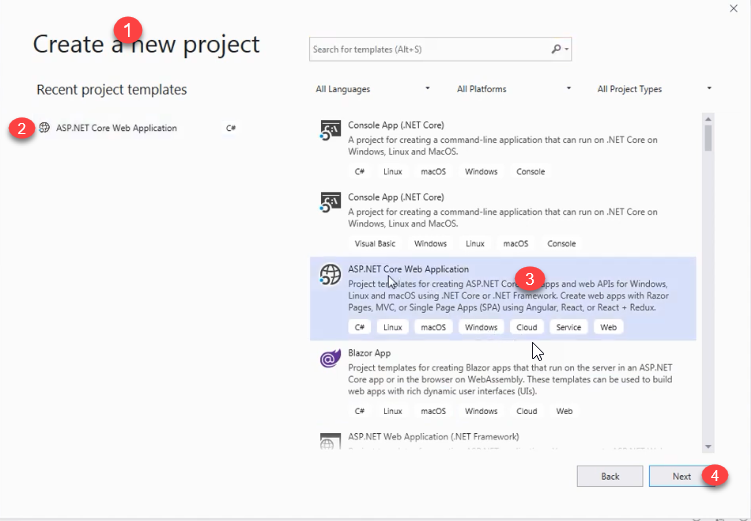
----------------------------------

ابتدينا الشغل النضيف شغل الويب API بقه و جماله

اول حاجه اذاى اعمل مشروع كور   
طبعا احسن حاجه تحملوا Visual Studio 2019 و انتوا بتسطبوه بقه تختاروا الاختيارات دى عشان الكور يشتغل معاكوا و يبقى كده انتوا حملتوا ال Net CORE SDK من خلال التسطيب يا رجاله   


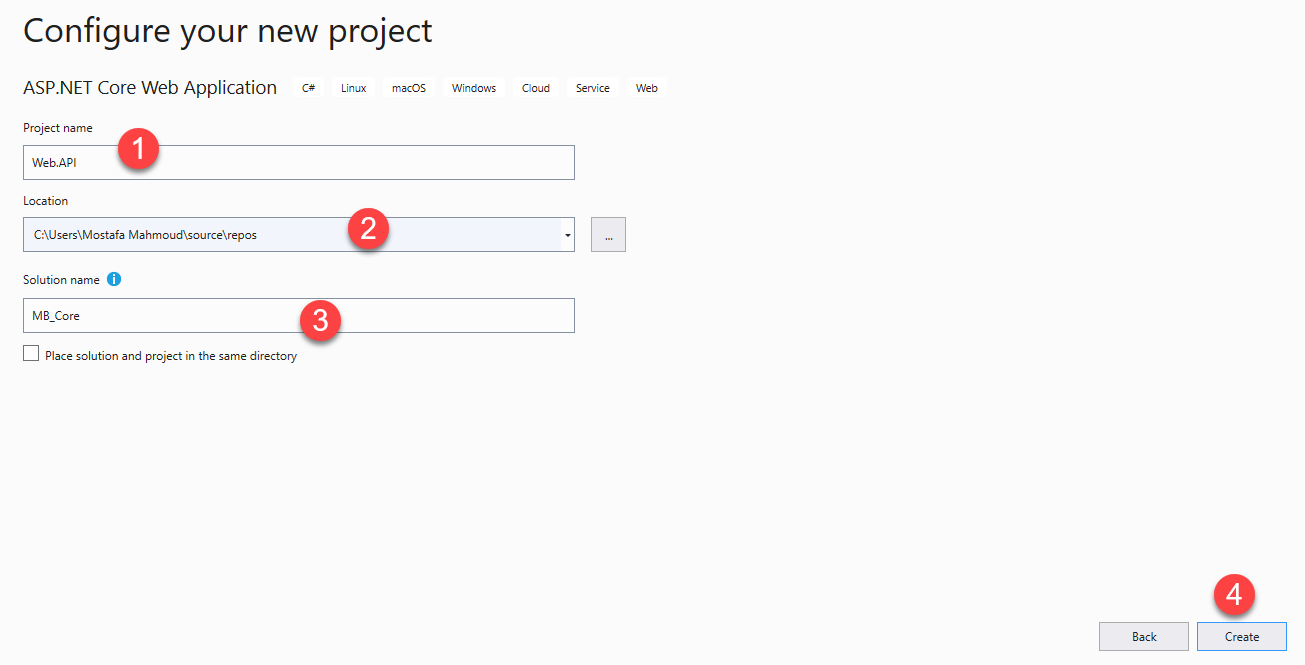
فى حاله انك سطبت و مش عاجبك ال .NET Core SDK عشان قدم شويه و عاوز حاجه احدث فانت كل الى هتعمله انك تخش على اللينك ده و تحمل ال SDK عادى خالص   
<https://dotnet.microsoft.com/download>  


و لما اجى اكريت بقه بروجيكت كور اتبع الكلام ده



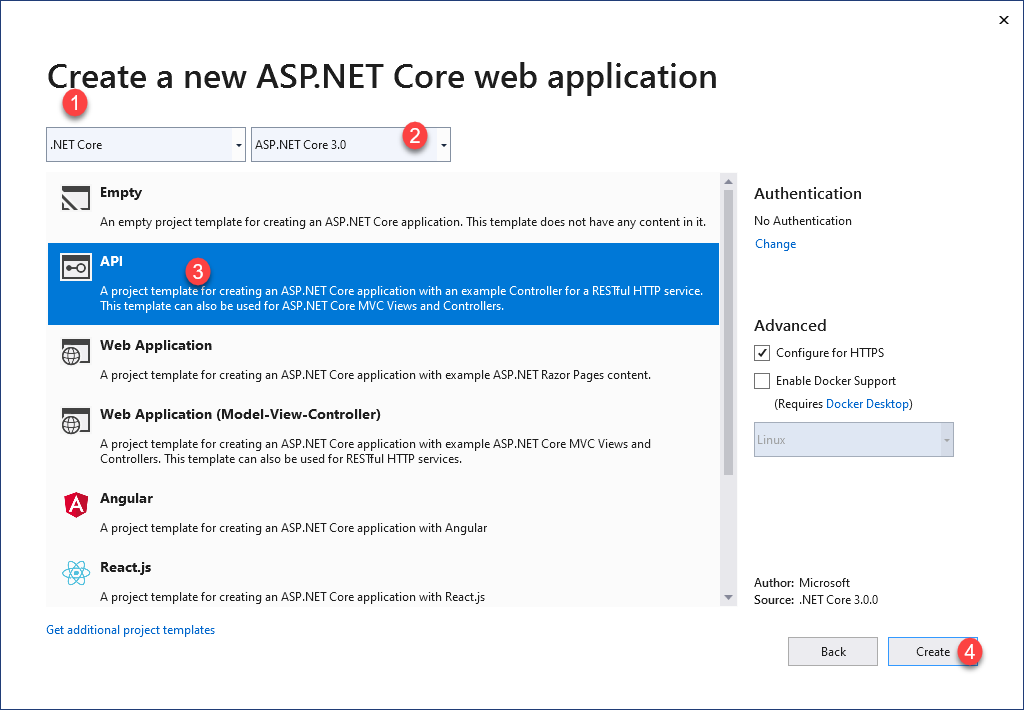
و ادى البروجيكت اسم بقه و اقوله Create   
و بعد كده بنحتار هل هو هيبقى API , MVC ولا ايه بالظبط هتلاقى كذا تيمبليت بس انا عن نفسى بحتار بين الاتنين دول بس و ممكن تختار wep application عادى

بس عشان هنشرح بالتفصيل فيفضل بقه انك تختار Empty و تشكل انت المشروع بايدك بقه



هنسمى المشروع بالاسم الى انت شايفه ده عشان ده هيبقى خاص بالحاجات ال API و طبعا ده هيبقى واحد من ضمن كذا مشروع عشان كده هعمل سوليوشن باسم مختلف و يبقى كل المشاريع تحت سليوشن واحد

هختار ال .net core و هخطار **اصدار ال 3** عشان ده الى مدعوم هختار المشرع ال Empty عشان هنشرح **انما لو فى الحاله الطبيعيه هبقى عاوز انجز فهختار ال API**

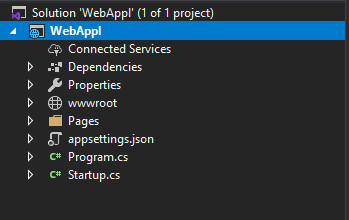


################################################################

تعالى نشرح فايلات المشروع بقه

-----------------

خلى بالك انا مسحت المشروع الى انا عملته MVC و ابتديت واحد جديد wep application عشان اوريك اذاى هخليه ذى بتاع ال MVC بالتدريج فتهم الدنيا اكتر تعالى بقه نشرح فايلات المشروع ال Empty ده



اول حاجه لو دوست على Wep.APl هيفتح معايا Wep.APl.csproj ملف المشروع من الاخر و ده بيبقى فيه كل فولدرات البروجيكت و الريفرنس و الباكيدجات الى انا حملتها تفدر تقول عليها عامل ذى ال package.json فى الانجيولار

**Dependencies** دى ذى ال References و دى هنا متقسمه ل 4 حاجات

1. Analysis : و دى بتبقى فيها الريفرانس الى ليه علاقه بالانالسيس
2. تانى حاجه اframeworks و دى الملفات الى ليها علاقه باى فريم ورك هتستخدمه فى المشروع
3. Nuget دى بقه اى حاجه متسطبه من على ال Nuget
4. SDK دى اى حاجه نزلت مع ال SDK

**Proparities** دى جواها فايل اسمه launchSetting.json بيحتوى على اعدادات المشروع و اعدادات الديباجينج و ال URL بتاعه و كده و البورتات الخ الخ ممكن تعمل ايديت فيه و تعمل حاجه على مزاجك عادى

**Wwwroot** دى بقه جواها فولدرات الجافا سكريبت و ال image و css دى تعتبر ال Assets بتاعت المشروع و غالبا فى ال API مش هستخدمها الا للضروره القصوى

**Appsettings** ده بقه ذى ما تقول كده ده ال wep config بشكله الجديد هههههههههه يعنى تحط فيه connection string و المشروع كله هيبقى شايفه و ده عشان يوفر عليا انى اعد اكتب الكونيكشن سترينج مليون مره فى الموقع و لو حبيت اغيره اغيره فى المكان ده بس و ابقى مش مضطر اغيره فى مليون مكان و ده بقه تحط فيه اى حاجه مهمه سواء خاصه بالابليكيشن او السيرفر الى هيترفع عليه المشروع

**Program.cs** ده بقه اول ملف بيشتغل فى المشروع من الاخر ده الى بيقوم المشروع و جواه كود هنشرحله دلوقتى

namespace Web.API

{

public class Program

{

public static void Main(string[] args)

{

CreateHostBuilder(args).Build().Run();

}

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

Host.CreateDefaultBuilder(args)

.ConfigureWebHostDefaults(webBuilder =>

{

webBuilder.UseStartup<Startup>();

});

}

}

هنا يا سيدى بقوله اول ما تعمل run للبرنامج روح اعملى هوست و بعد ما تعمل هوست خلاص شغل البروجيكت بالهوست

ده و ده بيرنه على حسب ما انت محدد يعدى لو هوست دوكر او اباتشى او اى اى اس هيرن بيهم عادى على حسب الكونفيجريشن الى انت عاملها عن طريق انه بيشغل كلاس ال Startup و ده اصلا كلاس موجوده فى الفايل الى تحته على طول الى اسمه startup

**Startup.cs** : ده بقه عامل ذى ال **appModule** الى فى انجيولار فهتلاقى ان فى اتنين فانكشن

اولهم دى

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

**الفانكشن دى اى ميدل وير هتعملها ذى مثلا حوار ال auth بتيجى تشغلها جوه الفانكشن دى**

**اذا Configure => middleware**

الميدل وير دى عباره عن فانكشن الكود بيناديها كل شويه يعنى حاجه وسيطه ففى الغالب دى بتتعمل لما يكون انت عاوز حاجه كل اما تيجى تعمل ريكويست يحصل تغيير عليها او يحصل فحص ليها و هناخد امثله ان شاء الله قدام

اما الفانكشن التانيه

**ConfigureServices**(IServiceCollection services)

دى بقه يا سيدى بتضيف فيها اى سيرفيس خاصه بيك عشان تعرف تشغلها Dependency injection

**ConfigureServices =>IOContainer for DI to Services**

**و ادى الكود بتعها يفهمك يا اسطى الدنيا شويه برضوا**

namespace Web.API

{

public class Startup

{

public Startup(IConfiguration configuration)

{

Configuration = configuration;

}

public IConfiguration Configuration { get; }

// This method gets called by the runtime. Use this method to add services to the container.

public void ConfigureServices(IServiceCollection services)

{

services.AddControllers();

}

// This method gets called by the runtime. Use this method to configure the HTTP request pipeline.

public void Configure(IApplicationBuilder app, IWebHostEnvironment env)

{

if (env.IsDevelopment())

{

app.UseDeveloperExceptionPage();

}

app.UseHttpsRedirection();

app.UseRouting();

app.UseAuthorization();

app.UseEndpoints(endpoints =>

{

endpoints.MapControllers();

});

}

}

}

الفايل ده هنشتغل فيه كتير و هتفهمه و هتبقى ذى الفل فمتشغلش دماغك بيه دلوقتى   
عشان تبقى فاهم بقه السايكل بتحصل اذاى   
أولا انت اول ما بتشغل المشروع بيروح يفتح الفانكشن الى اسمها Main فى الفايل program.cs لان دى الفانكشن الى بيبتدى بيها اى مشروع و بعد ما بيدخل فيها بيروح منادى على الفانكشن الى اسمها StartUp نفس الفايل   
و دى بتروح مشغله البروجيكت خلاص بالهوست   
أول ما المشروع خلاص هيشتغل بيروح رايح أول حاجه على اتنين فانكشن مهمين جدا   
الاولى الى هي بتعمل Register لكل السيرفيسيس الخاصه بيك الى هى اسمها ConfigureServices

بعديها على طول بيروح ينادى للفانكشن الى فيها ال PipeLines , Middleware و الى هى اسمها Configure  
و الى بيخشلها اتنين فاريبول الاول خاص بالابليكيشن و التانى خاص ببيئه العمل

MB

################################################################

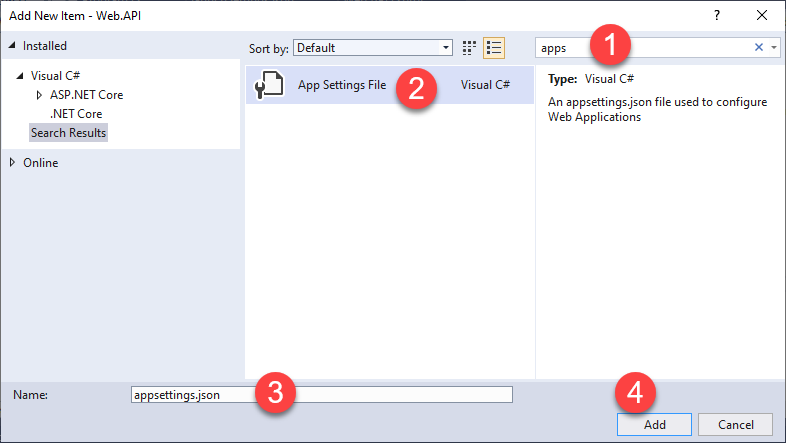
AppSetting.json

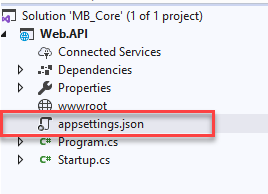
---------------------------------------------------------------

**Appsettings** ده بقه ذى ما تقول كده ده ال wep config بشكله الجديد هههههههههه يعنى تحط فيه connection string و المشروع كله هيبقى شايفه و ده عشان يوفر عليا انى اعد اكتب الكونيكشن سترينج مليون مره فى الموقع و لو حبيت اغيره اغيره فى المكان ده بس و ابقى مش مضطر اغيره فى مليون مكان و ده بقه تحط فيه اى حاجه مهمه سواء خاصه بالابليكيشن او السيرفر الى هيترفع عليه المشروع

و بيقى كتابته Json بالشكل ده

و دى بحطها تحت البروجيكت على طول يعنى فى اول ليفل جوه فولدر المشروه جنب Startup.cs , program.cs





{

"Logging": {

"LogLevel": {

"Default": "Information",

"Microsoft": "Warning",

"Microsoft.Hosting.Lifetime": "Information"

}

},

"AllowedHosts": "\*",

"ConnectionString": "My Connection String Info",

"ProjectName" : {

"DefaultName": "Core Project",

"RealName" : "Badawy Project"

}

}

طيب انا كتبت الكلام ده فى ال AppSettings.json عاوز اقراه بقه من اى مكان بقه اعمل ايه ؟؟

بص يا سيدى انا هقراه فى ال Startup.cs فتعالى اوريك هيتقرأ اذاى

namespace Web.API

{

public class Startup

{

public string ConnString { set; get; }

public Startup(IConfiguration configuration)

{

Configuration = configuration;

}

public IConfiguration Configuration { get; }

// This method gets called by the runtime. Use this method to add services to the container.

public void ConfigureServices(IServiceCollection services)

{

services.AddControllers();

}

// This method gets called by the runtime. Use this method to configure the HTTP request pipeline.

public void Configure(IApplicationBuilder app, IWebHostEnvironment env)

{

if (env.IsDevelopment())

{

app.UseDeveloperExceptionPage();

}

app.UseHttpsRedirection();

app.UseRouting();

app.UseAuthorization();

app.UseEndpoints(endpoints =>

{

endpoints.MapControllers();

});

app.Run(async (context) =>

{

var ProjectName = Configuration["ProjectName:RealName"];

ConnString = Configuration["ConnectionString"];

await context.Response.WriteAsync(ProjectName + " -- " + ConnString);

});

}

}

}

ذى ما انت شايق يا كبير عشان تقرأ حاجه فى ال AppSettings بتروح رايح على الكلاس الى انت هتقرا فيه من ال AppSettings و عامل Dependency Injection من نوع IConfigration و كده يا كبير انت بقى سهل تقرأ من ال AppSettings اى حاجه فيها و تقدر طبعا تسجلها فى اى فاريبول او بروبارتى او اى فيو داتا و تعرضها فى اى حته او تستخدمها باى شكل بقه براحتك

كده اول ما هتعمل Run هيروح يكتب فى البراوزار اسم المشروع و الكونيكشن سترينج بتاعه

ملحوظه و انت بتقرأ اى حاجه مش مباشره من ال AppSetting هتعمل بين كل parent , Child نقطتين فوق بعض مش دوت ذى ما انت شايف فى المثال ده

var ProjectName = Configuration["ProjectName**:**RealName"];

MB

################################################################

Another Way to Read from AppSetting.json

---------------------------------------------------------------

الطريقه دى انى اشوف ال Json Object و اعمل Classes تخلينى اعرف اقرأ منها فى اوبجيكت عادى و

طيب دلوقتى انا عملت ال Json كده

{

"ConnectionString": {

"DefaultConnStr": "My Connection String Info"

},

"ProjectName": {

"DefaultName": "Core Project",

"RealName": "Badawy Project"

}

}

بالتالى انا دلوقتى هعمل كلاس اسميه ProjectName و كلاس تانى اسمه connectionString و هيبقى شكله كده

namespace Web.API.Settings

{

public class ProjectName

{

public string DefaultName { get; set; }

public string RealName { get; set; }

}

public class ConnectionString

{

public string DefaultConnStr { get; set; }

}

}

كده ياباشا داخل فايل Settings.cs فيه اتنين classes الاول بتاع ال ConnectionString و التانى بتاع ال ProjectName و لازم تبقى عارف ان اى حاجه اساسيه فى ال Json و بيتعلمها كلاس او هى معموله كده لوحدها و مفيش تحتها اى حاجه فدى تعتبر Section و ده لانها تحت ال root مباشر

فلما اجى اقراهم بقه هروح اقوله هاتلى السيسكشن الى اسمه بروجيكت نيم و السكشن الى اسمه كونيكشن سترينج و هيتكتب الكلام ده بالطريقه دى

namespace Web.API

{

public class Startup

{

public string DefaultConnString { set; get; }

public string RealName { set; get; }

public string DefaultName { set; get; }

public Startup(IConfiguration configuration)

{

Configuration = configuration;

}

public IConfiguration Configuration { get; }

// This method gets called by the runtime. Use this method to add services to the container.

public void ConfigureServices(IServiceCollection services)

{

services.AddControllers();

}

// This method gets called by the runtime. Use this method to configure the HTTP request pipeline.

public void Configure(IApplicationBuilder app, IWebHostEnvironment env)

{

if (env.IsDevelopment())

{

app.UseDeveloperExceptionPage();

}

app.UseHttpsRedirection();

app.UseRouting();

app.UseAuthorization();

app.UseEndpoints(endpoints =>

{

endpoints.MapControllers();

});

app.Run(async (context) =>

{

var ConString = Configuration.GetSection("ConnectionString").Get<ConnectionString>();

DefaultConnString = ConString.DefaultConnStr;

var ProjectName = Configuration.GetSection("ProjectName").Get<ProjectName>();

RealName = ProjectName.RealName;

DefaultName = ProjectName.DefaultName;

await context.Response.WriteAsync(ProjectName.RealName + " -- " + DefaultConnString);

});

}

}

}

طيب قى الغالب بقه يا سيدى انت بتبقى محتاج الكونيكشان سترينج فى ال Startup.cs عشان ياخد الكونيكشن فى اول البروجيكت كده :D

MB

################################################################

اذاى اظهر الاكسيبشن فى البروداكشن

------------------------------

بص يا سيدى انت هنا فى الكود بتقوله لو الانفيرومينت هى الديفلوب مينت فانت جبلى

app.UseDeveloperExceptionPage();

غير كده استخدم ال Hsts و دى بتفتح البيدج العاديه خالص بتاعت ضرب اى موقع لما حاجه تبوظ فيه   
طبعا انا حطيت فى ال Run كود throw new exception عشان يطلعلى ايرور اجرب بيه ههههههههههههه انما عمرى ما هكتب كده فى الحقيقه اصلا ههههههههههههه المكان ده يبقى لكونيكشن استرينج لحاجه مهمه كده و محترمه و بنت ناس مش اى حاجه و السلام

public void Configure(IApplicationBuilder app, IhostingEnvironment env)

{

if (env.IsDevelopment())

{

app.UseDeveloperExceptionPage();

}

else

{

app.UseHsts();

}

app.Run(async (context) =>

{

throw new Exception();

});

}

و دى انواع بيئات المشروع

namespace Microsoft.AspNetCore.Hosting

{

public static class HostingEnvironmentExtensions

{

public static bool IsDevelopment(this IHostingEnvironment hostingEnvironment);

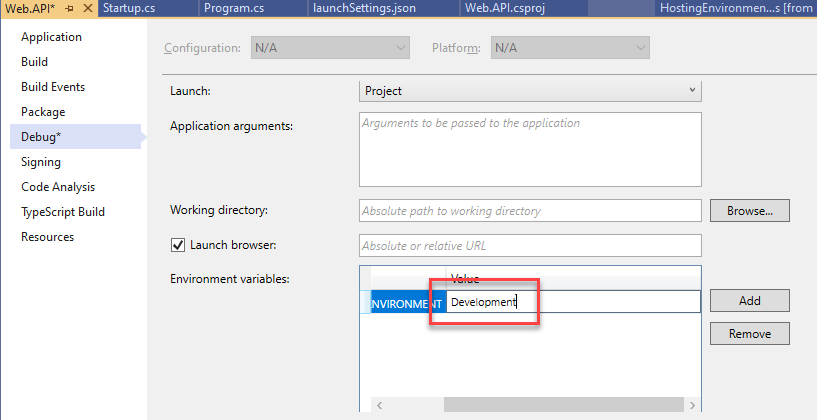
public static bool IsEnvironment(this IHostingEnvironment hostingEnvironment, string environmentName);

public static bool IsProduction(this IHostingEnvironment hostingEnvironment);

public static bool IsStaging(this IHostingEnvironment hostingEnvironment);

}

}

و دى اذاى بغيره فى بروبرتى المشروع   
  
خد بقه افهملك شويه حاجه حلوه من الموقع ده

<https://docs.microsoft.com/en-us/aspnet/core/tutorials/first-web-api?view=aspnetcore-3.1&tabs=visual-studio>

MB

################################################################

AddMvc( ) and UseMvc() and AddControllers()

---------------------------------------------------------------

دى يا سيدى فانكشنز بعملها اضافه فى ال Services عشان اقدر استخدم ال MVC pattern و عن طريقها اقدر اعمل Controller و اكلمه بقه فى ال API لان ذى ماانت عارف ال MVC عباره عن ديزاين باترن مكون من   
Model-View-Controller

فعشان الديزاين باترن دى تشتغل و اقدر اهندل فى الكود انه يعرف يقرأ و يظبط و يربط الفايلات بطريقه ال MVC مع بعض فبستخدم السيرفيس دى

public void ConfigureServices(IServiceCollection services)

{

**services.AddMvc();** **// دى بنستخدمها مع الكور 2 و 3**

}

طيب دلوقتى احنا ضيفناها عاوزين نستخدمها بقه يبقى المفروض اقوله ياباشا انا عاوز مع كل ريكويست استخدم نظام ال MVC ده يبقى ياباشا عليك و على ال Middleware روح قوله فى الكونفيجريشن انك عاوز تستخدم الميدل وير بتاعت ال MVC

public void Configure(IApplicationBuilder app, IWebHostEnvironment env)

{

if (env.IsDevelopment())

{

app.UseDeveloperExceptionPage();

}

app.UseHttpsRedirection();

app.UseRouting();

app.UseAuthorization();

app.UseEndpoints(endpoints =>

{

endpoints.MapControllers();

});

**app.UseMvc();**

}

طيب فى حاجه احسن بقه انك تستخدم ال AddController دى ذى الاتنين الى فاتوا مع بعض و احسن منهم و مجرد بتقوله فى السيرفيس انك عاوز تضيف للمشروع خدمه اضافه الكنترولار و بعد كده بتروح تكلم الكنترولارس عادى خالص

public void ConfigureServices(IServiceCollection services)

{

services.AddControllers(); **// دى مخصصه لل كور 3**

}

**هنا انت مش محتاج تظبط حاجه فى الكونفيجريشن خلاص**

public void Configure(IApplicationBuilder app, IWebHostEnvironment env)

{

if (env.IsDevelopment())

{

app.UseDeveloperExceptionPage();

}

app.UseHttpsRedirection();

app.UseRouting();

app.UseAuthorization();

app.UseEndpoints(endpoints =>

{

endpoints.MapControllers();

});

}

**بص من الاخر كده الزتونه انك لو هتستخدم Controller فى اى مشروع لازم تستخدم السيرفيس و الكونفيجريشن بتوع ال MVC الى هما addMvc , UseMvc**

**فى الاصدار التالت للكور بنستخدم**

**services.AddControllers();**

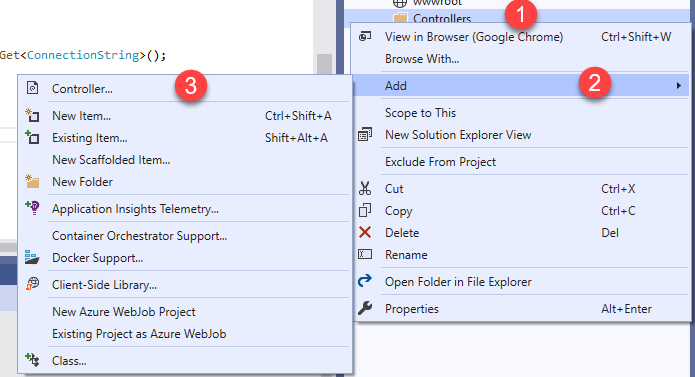
MB

################################################################

controller

------------------

بعد ما شغلنا ال MVC او ال AddController فى المشروع كده خلاص جه الوقت الى خلاص اقدر استخدم بقه ال controller فى المشروع و ذى ما انت عارف الكونترولر ده بيبقى فيه كل حاجه خاصه بال APIs و اذاى اعمل Route, HttpGet , HttpPost,HttpDelete , HttpPut عشان اكلم الداتا بيز بيهم و اعمل ريكويستات بتكلم السيرفر و ترجع منه داتا او تضيفله او تعدل عليه او تمسحه   
و ادى ذى ما انت شايف عملنا فولدر للكنترولار و هنبتدى نعمل كنترولار اسمه cities اعلمك عليه



اول حاجه اننا بنعمل كلاس الكونترولار بالاسم ده CitiesController

بعد كده احنا المفروض بنعمل انهيرتانس للكلاس ده من كلاس ال ControllerBase و ده لان الكلاس ده بيحتوى على كميه فانكشنز كتير جدا ليها علاقه بشغل ال APIs و كده   
تانى حاجه المفروض يبقى فيه RouterPrefix للكلاس ده عشان مش كل شويه اكرره فى كتابه الروتر و ده بيبقى اتربيوت فوق الكلاس و بما اننا شغالين كور فهيبقى المسميات مختلفه شويه عن المسميات بتاعت الويب APi العادى هنا الروتربريفيكس بيتكتب كده

//[RoutePrefix("api/Cities")] // old version for Web API

//[Route("api/Cities")] // New Version For Core API

[Route("api/[controller]")] // New Version For Core API

[ApiController]

الاتنين اتربيوتس دول بيبقوا فوق الكلاس عشان يدوله خواص معينه يعنى الروت هيخلى ان اى فانكشن تابعه للكلاس دى بتاخد الروت الثابت الى فوق ده قبل الروت بتاعها   
اما ال apiController ده بيدى خواص ال Api للكلاس   
اول فانكشن بقه هنا هى ال GET و هى الى بجيب بيها الداتا و دى هكتبها كده

// GET: api/Cities

[HttpGet]

public List<Object> Get()

{

return new List<Object>() {

new {id=1 , Name = "Mahmoud Badawy"},

new {id=2 , Name = "Mostafa Badawy"}

};

}

و ذى ما انت شايف هى واخده اتربيوت اسمه [HttpGet] فوقها و دى يا صديقى لان الاتربيوت ده بيتيج ليها انها تعمل HttpRequest من نوع Get الى هو بيجيب الداتا و يرجعهالى   
طيب انا عشان اعرف اكلم ال API دى المفروض يبقى عندى لينك و ده بكونه اذاى ؟؟  
بص يا سيدى ده بيبقى Domain + api/ControllerName/[CustomRouter---Optional]

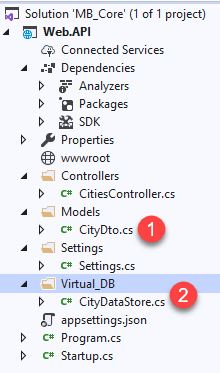
يعنى فى حالتنا دى احنا عشان نكلم الفانكشن دى هيبقى اللينك كده   
<https://localhost:44378/api/Cities>

الدومين الى انا عامل عليه اصفر ده انا جايبه من الموقع لما شغلته و طلع لينك فوق   
اما الباقى فده من ال Route او RouterPrefix

MB

################################################################

Models

------------------

بص المودل ده فولدر بحط فيه الكلاسات الى انا هستخدمها للتعامل مع الداتا بيز بتعتى فهنا انا هعمل الفولدر ده و احط فيه كلاس اسمه CityDto ده الى هيبقى فيه البروبرتىس الخاصه بالمدينه

namespace Web.API.Models

{

public class CityDto

{

public int Id { get; set; }

public string Name { get; set; }

public string Description { get; set; }

}

}

ذى ما انت شايف الكلاس فوق اهه اتعمل للموديل كده المفروض بقه هعمل كلاس تاني يبقى بديل للداتا بيز بما اننا لسه مشتغلناش على الداتا بيز و دى الى هحط فيها المعلومات الى هتعامل معاها اعتبرها دى الداتا بيز بتعتك من الاخر و انا حاطط الكلاس دى فى فولدر اسمه Virual\_DB فى فايل اسمه CityDataStore.cs

using Web.API.Models;

namespace Web.API.Virtual\_DB

{

public class CityDataStore

{

public static CityDataStore CurrentCities { get; } = new CityDataStore();

public List<CityDto> Cities { get; set; }

public CityDataStore()

{

Cities = new List<CityDto>()

{

new CityDto()

{

Id=1,

Name="Mahmoud Badawy",

Description="Mahmoud Badawy Description"

},

new CityDto()

{

Id=2,

Name="Mostafa Badawy",

Description="Mostafa Badawy Description"

},

new CityDto()

{

Id=3,

Name="yassen Badawy",

Description="yassen Badawy Description"

},

new CityDto()

{

Id=4,

Name="Noha Badawy",

Description="Noha Badawy Description"

},

};

}

}

}

كده انت كانك عامل داتا بيز بالظبط و دى ال dummy data  
المهم دلوقتى بقه المفروض اروح اقرأ الكلام ده و اشتغل عليه بقه من ال Controller ياباشا عشان نجرب بقه بالبوست مان و نهيص

namespace Web.API.Controllers

{

//[RoutePrefix("api/Cities")] // old version for Web API

//[Route("api/[controller]")] // New Version For Core API

[Route("api/Cities")] // New Version For Core API

[ApiController]

public class CitiesController : ControllerBase

{

// GET: api/Cities

[HttpGet]

public IActionResult Get()

{

try

{

List<CityDto> Data = CityDataStore.CurrentCities.Cities.ToList();

if (Data == null)

{

return NoContent();

}

return Ok(Data);

}

catch (Exception)

{

return BadRequest();

throw;

}

}

// GET: api/Cities/1

[HttpGet("{id}")]

public IActionResult Get(int id)

{

try

{

CityDto Data = CityDataStore.CurrentCities.Cities.Where(x => x.Id == id).FirstOrDefault();

if (Data == null)

{

return NotFound();

}

return Ok(Data);

}

catch (Exception)

{

return BadRequest();

throw;

}

}

// POST: api/Cities

[HttpPost]

public IActionResult Post(CityDto VM)

{

try

{

CityDto item = new CityDto();

item.Id = CityDataStore.CurrentCities.Cities.Select(p => p.Id).DefaultIfEmpty(0).Max() + 1;

item.Name = VM.Name;

item.Description = VM.Description;

CityDataStore.CurrentCities.Cities.Add(item);

var Data = CityDataStore.CurrentCities.Cities;

return Ok(Data);

}

catch (Exception )

{

return BadRequest();

throw;

}

}

// PUT: api/Cities/5

[HttpPut("{id}")]

public IActionResult Put(int id, CityDto VM)

{

try

{

CityDto item = CityDataStore.CurrentCities.Cities.Where(x => x.Id == id).FirstOrDefault();

if (item == null)

{

return NotFound();

}

item.Name = VM.Name;

item.Description = VM.Description;

var Data = CityDataStore.CurrentCities.Cities;

return Ok(Data);

}

catch (Exception)

{

return BadRequest();

throw;

}

}

// DELETE: api/Cities/5

[HttpDelete("{id}")]

public IActionResult Delete(int id)

{

try

{

CityDto item = CityDataStore.CurrentCities.Cities.Where(x => x.Id == id).FirstOrDefault();

if (item == null)

{

return NotFound();

}

CityDataStore.CurrentCities.Cities.Remove(item);

var Data = CityDataStore.CurrentCities.Cities;

return Ok(Data);

}

catch (Exception)

{

return BadRequest();

throw;

}

}

}

}

دلوقتى ياباشا انا عملت الكونترولار و اول حاجه عملتاه الروتربريفيكس الى فوق الكلاس عشان ده يبقى عام لكل الكلاس بتاع الكنترولار ده و مكررهاش فى كل الروتس الى موجوده فيه

تانى حاجه انا عملت CRUD فكان اول حاجه ال GetList اجيب بيها كل الداتا الى فى ال cities

[HttpGet]

public IActionResult Get()

{

try

{

List<CityDto> Data = CityDataStore.CurrentCities.Cities.ToList();

if (Data == null)

{

return NoContent();

}

return Ok(Data);

}

catch (Exception)

{

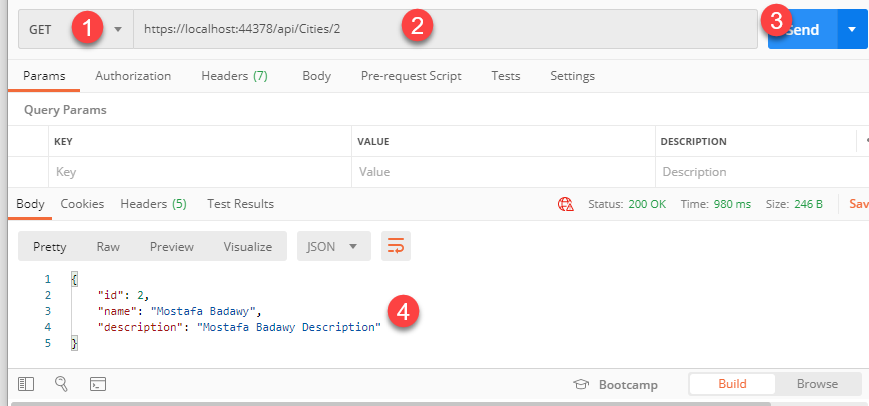
return BadRequest();

throw;

}

}

و ذى ما انت شايف انا بقوله اول حاجه فوق الفانكشن بقوله ان دى من نوع ال HttpGet و بعد كده بقوله هاتلى كل الداتا فى ليست و رجعهالى فى ال OK الى هى من نوع IActionResult و دى كلاس حديثه بترجع HttpResponse بشكل احسن و اوفر فى كتابه الكود … طيب فى حاله مفيش داتا بقوله رجعلى يا صديقى NoContent عشان اعرف ايه الحاله الحاليه لكل ريكويست بيحصل و الرد عليه اما فى حاله حصل مشكله من السيرفر او من الكود عمتا بقوله دخلنى فى ال الكاتش و رجع معايا BadRequest

فى بقه الجيت بتاعه OneItem دى يا صديقى بحدد فى ال Route بتعها و انا بعمل ال HttpGet انها هتبعت فى اللينك رقم ال id بتاع ال item الى هى عاوزه معلومات عنها بالتحديد   
و طريقه ارسال الريكويست بتعها على البوست مان هى هى نفس الى فاتت بس فرق اللينك مش اكتر   
و ادى الطريقه على ام oneItem هنا اهه   


اما فى حاله ال insert فالموضوع هيختلف بقه اولا انا هستخدم فوق الفانكشن HttpPost و هابعت براميتار فيه Item كامل عشان اضيفه و بضيفه على ال items القديمه و ارجع الكل عشان اطمن انه انضاف يا صديقى

// POST: api/Cities

[HttpPost]

public IActionResult Post(CityDto VM)

{

try

{

CityDto item = new CityDto();

item.Id = CityDataStore.CurrentCities.Cities.Select(p => p.Id).DefaultIfEmpty(0).Max() + 1;

item.Name = VM.Name;

item.Description = VM.Description;

CityDataStore.CurrentCities.Cities.Add(item);

var Data = CityDataStore.CurrentCities.Cities;

return Ok(Data);

}

catch (Exception )

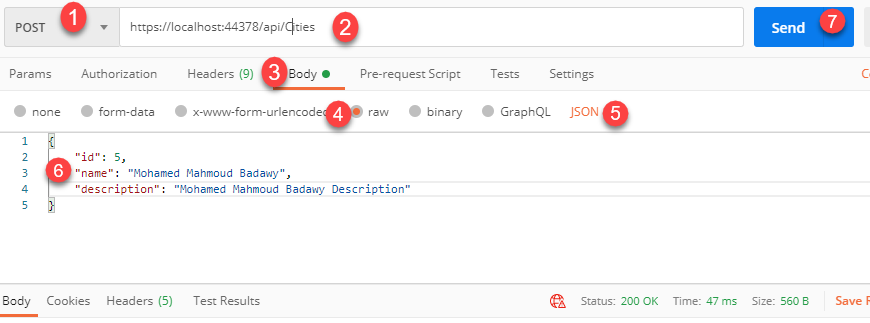
{

return BadRequest();

throw;

}

}

و طريقه استخدام البوست مان فى الجزئيه دى ان ببعت الاوبجيكت بالايتم الى انا محتاجه فى ال Body من نوه جيسون   


و التعديل و المسح نفس الحكايه باختلاف نوع ال HttpRequest مش اكتر

MB

##########################################################################

Return Children Resourses

----------------------------------

ده واضح من اسم العنوان دلوقتى لو انا عاوز ارجع حاجه تعتبر child للاوبجيكت الى انا برجع منه دلوقتى اعملها اذاى فمثلا عندنا دلوقتى ال object CityDto و المفروض هخلى فيه حاجه اسمها PointOfInterest دى عباره عن لسته اوبجيكتات داخل اوبجيكت السيتى فأولا انا هنشأ اول حاه كلاس ال PointOfInterestDto

namespace Web.API.Models

{

public class PointOfInterestDto

{

public int Id { get; set; }

public string Name { get; set; }

public string Description { get; set; }

}

}

بعد كده هروح اضيف منه لسته اوبجيكتات داخل كلاس ال CityDto

namespace Web.API.Models

{

public class CityDto

{

public int Id { get; set; }

public string Name { get; set; }

public string Description { get; set; }

public ICollection<PointOfInterestDto> PointOfInterest { get; set; } = new List<PointOfInterestDto>();

public int NumPointOfInterest

{

get

{

return PointOfInterest.Count;

}

}

}

}

و طبعا لانى عاوز اعرف كل مدينه معاها كام PointOfInterest فقولتله هاتلى NumPointOfInterest من الحسبه دى بدل ما اعملها فى الكود يعنى

طيب دلوقتى بقه احنا المفروض نروخ نضيف بيانات ال City و نضيف لكل واحده ال PointOfinterest بتعتها

namespace Web.API.Virtual\_DB

{

public class CityDataStore

{

public static CityDataStore CurrentCities { get; } = new CityDataStore();

public List<CityDto> Cities { get; set; }

public CityDataStore()

{

Cities = new List<CityDto>()

{

new CityDto()

{

Id=1,

Name="Mahmoud Badawy",

Description="Mahmoud Badawy Description",

PointOfInterest = new List<PointOfInterestDto>()

{

new PointOfInterestDto()

{

Id=1,

Name="Mostafa Mahmoud Badawy",

Description="Mostafa Mahmoud Badawy Description"

},

new PointOfInterestDto()

{

Id=2,

Name="Yassen Mahmoud Badawy",

Description="Yassen Mahmoud Badawy Description"

}

}

},

new CityDto()

{

Id=2,

Name="Mostafa Badawy",

Description="Mostafa Badawy Description"

},

new CityDto()

{

Id=3,

Name="yassen Badawy",

Description="yassen Badawy Description"

},

new CityDto()

{

Id=4,

Name="Noha Badawy",

Description="Noha Badawy Description",

PointOfInterest = new List<PointOfInterestDto>()

{

new PointOfInterestDto()

{

Id=1,

Name="Mostafa Mahmoud Badawy",

Description="Mostafa Mahmoud Badawy Description"

},

new PointOfInterestDto()

{

Id=2,

Name="Yassen Mahmoud Badawy",

Description="Yassen Mahmoud Badawy Description"

}

}

},

};

}

}

}

الباقى بقه اننا نروح نعمل كونترولار جديد للبوينت اوف انتريست لوحده

namespace Web.API.Controllers

{

[Route("api/Cities/{CityId}/PointOfInterest")]

[ApiController]

public class PointOfInterestController : ControllerBase

{

[HttpGet]

public IActionResult GetPointOfInterestByCityId(int CityId)

{

CityDto City = CityDataStore.CurrentCities.Cities.Where(x => x.Id == CityId).FirstOrDefault();

if(City == null)

{

return NotFound();

}

return Ok(City.PointOfInterest);

}

[HttpGet("{id}")]

public IActionResult GetPointOfInterestByPointOfInterestId(int CityId , int id)

{

CityDto City = CityDataStore.CurrentCities.Cities.Where(x => x.Id == id).FirstOrDefault();

if (City == null)

{

return NotFound();

}

PointOfInterestDto PointOfInterest = City.PointOfInterest.Where(x => x.Id == id).FirstOrDefault();

if (City == null)

{

return NotFound();

}

return Ok(PointOfInterest);

}

}

}

ذى ما انت شايف ال الروت هعمله ليه علاقه بال City عشان يبقى معروف ان ال PointOfInterest حاجه من ضمن الحاجات الموجود فى ابناء ال Cities  
[Route("api/Cities/{CityId}/PointOfInterest")]  
هنا دى معناها ان الروت لازم يبقى فيه ال CityId عشان يدخلك فى الكنترولار ده و يجيبلك ال PointOfinterest بتعتها   
و دول بقه اتنين فانكشن بيجيبلولك ال PointOfinterest مره بدلاله ال CityId و مره بال CityId و كمان ال POI\_ID

namespace Web.API.Controllers

{

[Route("api/Cities/{CityId}/PointOfInterest")]

[ApiController]

public class PointOfInterestController : ControllerBase

{

[HttpGet]

public IActionResult GetPointOfInterestByCityId(int CityId)

{

CityDto City = CityDataStore.CurrentCities.Cities.Where(x => x.Id == CityId).FirstOrDefault();

if(City == null)

{

return NotFound();

}

return Ok(City.PointOfInterest);

}

[HttpGet("{id}")]

public IActionResult GetPointOfInterestByPointOfInterestId(int CityId , int id)

{

CityDto City = CityDataStore.CurrentCities.Cities.Where(x => x.Id == id).FirstOrDefault();

if (City == null)

{

return NotFound();

}

PointOfInterestDto PointOfInterest = City.PointOfInterest.Where(x => x.Id == id).FirstOrDefault();

if (City == null)

{

return NotFound();

}

return Ok(PointOfInterest);

}

}

}

جرب بقه تعمل ال GET Request باللينكات دى و شوف ايه النتايج و هتفهم

<https://localhost:44378/api/Cities/2>  
<https://localhost:44378/api/Cities/1/PointOfInterest>

<https://localhost:44378/api/Cities/1/PointOfInterest/1>

MB

##########################################################################

Work with Serialize Settings

-------------------------------------------

دى يا سيدى عشان لو انت عاوز ترجع الداتا جيسون بس يكون بفورمات معين ذى ان مثلا كل ال property تبقى بتبدأ بحرف كابتل بدل سمول بتروح رايح على الجزئيه الى بتكتب فيها ال Services فى ال ConfigureServices و تكتب فيها الاوبشن الى انت عاوزه ذى ان انت عاوز كل الاوبجيكتات ترجع البروبيرتيس بتعتها اول حرف فيهم كابتل كده

public void ConfigureServices(IServiceCollection services)

{

services.AddMvc().AddJsonOptions(X=> {

if(X.SerializerSettings.ContractResolver != null) {

var CastedResolver = X.SerializerSettings.ContractResolver as DefaultContractResolver;

CastedResolver.NamingStrategy = null;

}

});

}

انا من رايي بقه الكلام ده ملوش لازمه ههههههههههه الا فى حالات قليله جدا

MB

##########################################################################

Formatters and Content Negotiation

-------------------------------------------

ده يا سيدى بعمله لما اكون عاوز احدد نوعيه ال object الى هعمل بيه input للسيرفر او Output منه فعلى سبيل المثال انا عاوز ارجع الاوبجيكت على شكل جيسون بروح اقوله خلى الريكويست يرجع جيسون و لو عاوز XML هقوله خليه يرجع XML و كذلك فى ال Input بحددله حاجات معينه برضوا

الاوت بوت بحدده فى media type : accept Header   
اما ال Input بحدده فى ال media Type : content-type Header  
تعالى بقه اوريك مثال ان عاوز ارجع حاجه على شكل XML بدل من Json فهروح برضوا لفايل ال startUp و اكتب الكود ده فى السيرفيسيس برضوا عمتا اى اوبشن بيتكتب فى ال Services بنسبه كبيره

public void ConfigureServices(IServiceCollection services)

{

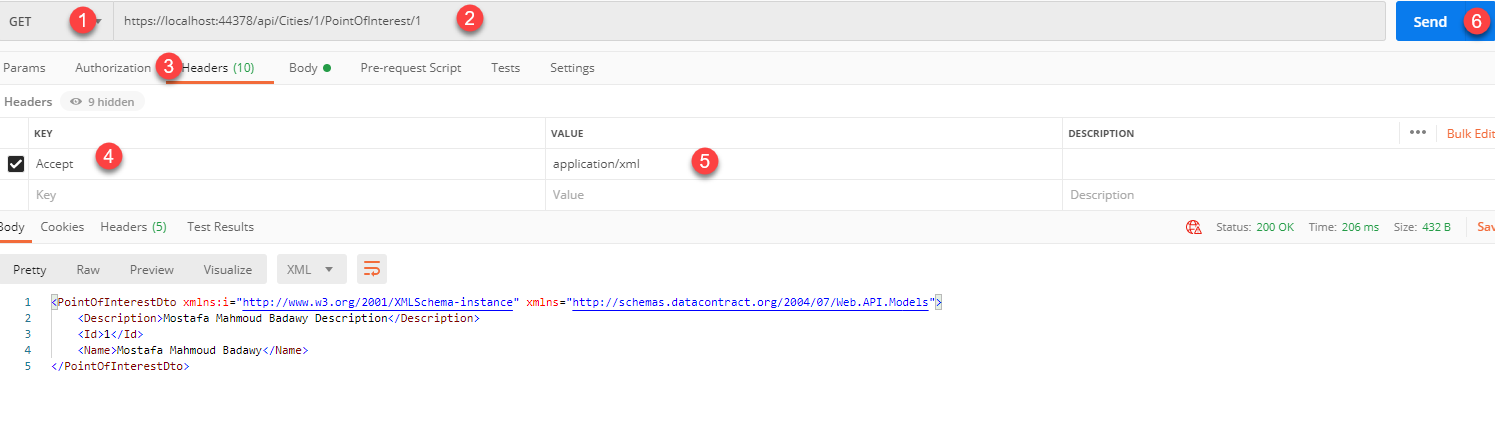
services.AddMvc().AddMvcOptions(X => {

X.OutputFormatters.Add(new XmlDataContractSerializerOutputFormatter());

});

}

فهتبقى تجربت الحاجه دى فى ال PostMan كده و ذى ما انت شايف اتا قولتله Accetp فى الهيدر ب application/xml يعنى الاوت بوت يرجع xml لو كتبله فاليو كمان content-type فى الهيدر ب application/xml هبقى ساعتها لازم ابعت الانبوت xml



MB

##########################################################################

Input Validation (Data Annotations and ModelState and Fluent Validation)

-------------------------------------------

دى يا سيدى انى اعمل الفاليديشن بتاعت الحاجه من الباك اند و دى طبعا حاجه جميله جدا جدا جدا   
بص يا سيدى اولا فى كذا طريقه الطريقه الاولى انى اعمل الفاليديشن ب If Conditions على الاوبجيكت الى جاى فى ال POST بس دى اسوأ طريقه   
الطريقه الاحسن بقه انى اعمل ال validation على ال Class الى انا عامله فى البراميتار بتاع الادخال بتاع البوست بال attributes الخاصه ب DataAnotations ذى المثال ده مثلا

using System.ComponentModel.DataAnnotations;

namespace Web.API.Models

{

public class CityDto

{

[Key]

public int Id { get; set; }

[Required(AllowEmptyStrings =false , ErrorMessage = "Name is Required")]

[MaxLength(50,ErrorMessage = "Max length for name is 50 Digit")]

public string Name { get; set; }

[MaxLength(100, ErrorMessage = "Max length for Description is 100 Digit")]

public string Description { get; set; }

public ICollection<PointOfInterestDto> PointOfInterest { get; set; } = new List<PointOfInterestDto>();

public int NumPointOfInterest

{

get

{

return PointOfInterest.Count;

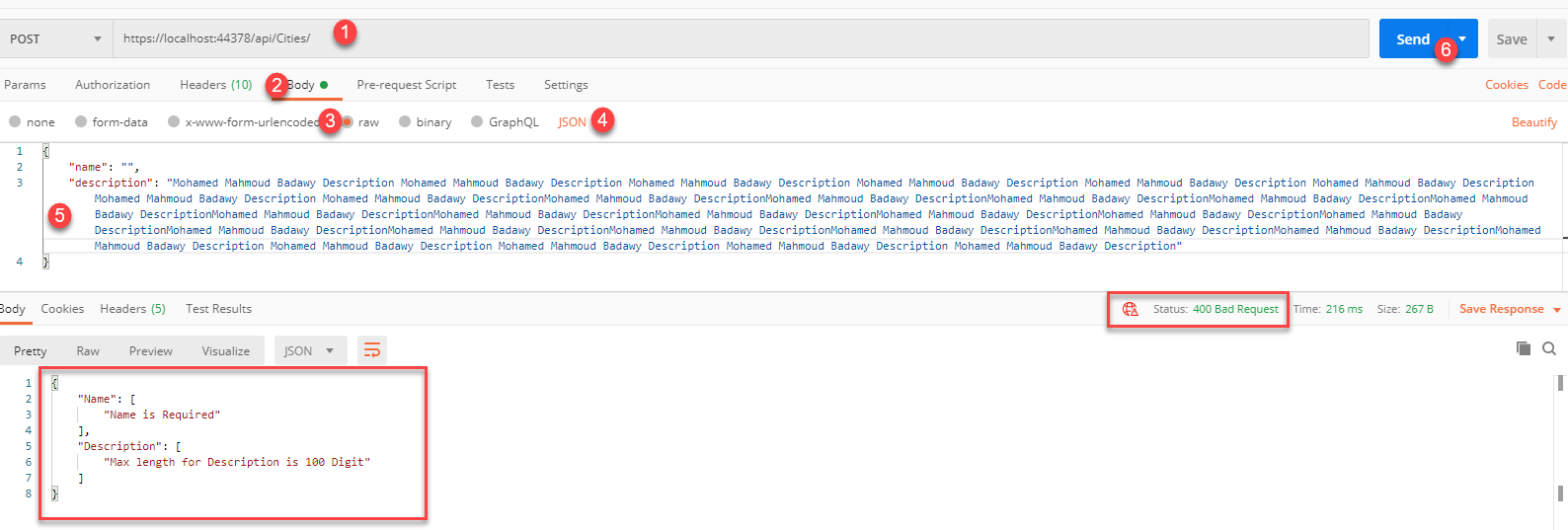
}

}

}

}

تعالى نجرب بقه بوست بايظ هههههههههههه



طب لو حاجه كومبليكس ذى ان مثلا ان انا عاوزه ان مينفعش يجيب ال Name بيساوى ال Description فى القيمه دى بقه بتتعمل فى الكود عن طريقه اضافه ال ModelState للايرور و دى بتتعمل كده

// POST: api/Cities

[HttpPost]

public IActionResult Post(CityDto VM)

{

try

{

if(VM.Name == VM.Description)

{

ModelState.AddModelError(

"Description",

"The Provided Description must be different from Name"

);

}

if (!ModelState.IsValid)

{

return BadRequest(ModelState);

}

CityDto item = new CityDto();

item.Id = CityDataStore.CurrentCities.Cities.Select(p => p.Id).DefaultIfEmpty(0).Max() + 1;

item.Name = VM.Name;

item.Description = VM.Description;

CityDataStore.CurrentCities.Cities.Add(item);

var Data = CityDataStore.CurrentCities.Cities;

return Ok(Data);

}

catch (Exception )

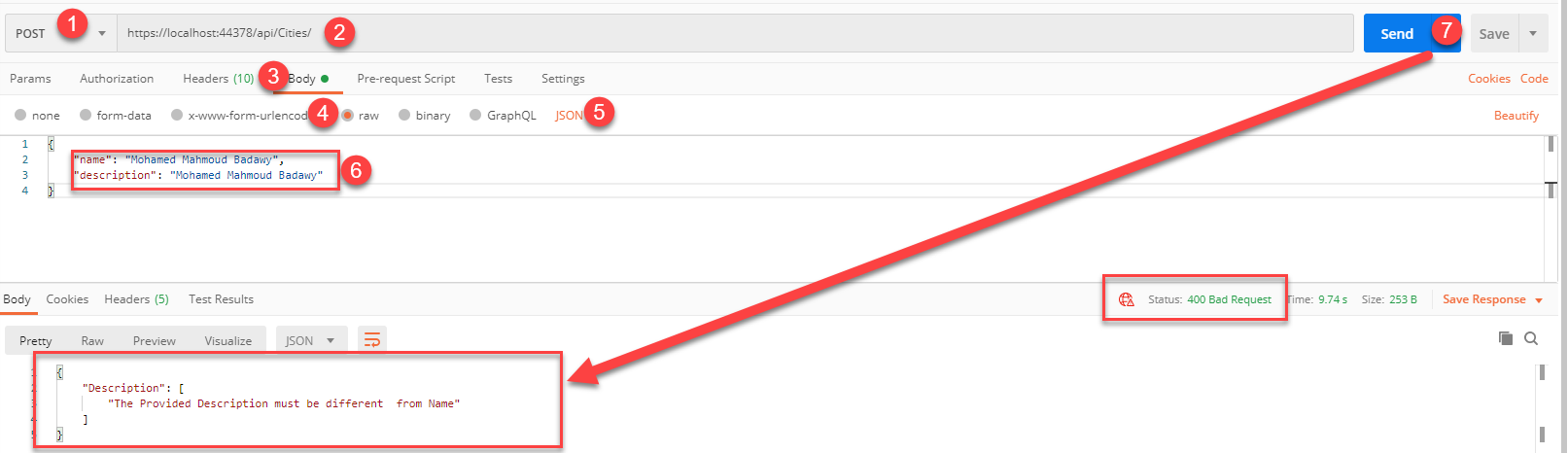
{

return BadRequest();

throw;

}

}



بيقولك بقه احسن حاجه و انضف حاجه فى الموضوع ده بقه حاجه اسمها FluentValidation ذى ال FluentAPI فشيخه بتكتب كل حاجه فى الكود بتاع الكونفيجريشن الخاص بالكلاس او الجدول

و دى باكيديج جميله جدا متوفره لل ASP.NET MVC , Web API , CORE و دى الريبو بتعتها و الدوكنتيشن و الشرح و كل حاجه

<https://github.com/FluentValidation/FluentValidation>

و دى الحاجه الخاصه بالكور بس نصيحه لفلك لفه فى الدوكمنيتشن كلها احسن من اول الجيت هب للموقع ككل هتستفاد اكتر

<https://docs.fluentvalidation.net/en/latest/aspnet.html>

MB

##########################################################################

Update By Put VS Patch

-------------------------------------------

لو عملت Update بال Put فسعتها لازم ابعت كل الاوبجيكت بالقيم الجديده و الا الى انا مبعتهوش هيبقى ب Null فى الداتا بيز لانى وديته فاضى

طيب لو عملت بال Patch ساعتها هو هيروح للحاجه الى هو عامل فيها update بس و يغيرها و خلاص من الاخر   
ال put هو عباره عن Full Update بينما ال Patch ده Particial update

// PUT: api/Cities/5

[HttpPut("{id}")]

public IActionResult Put(int id, CityDto VM)

{

try

{

CityDto item = CityDataStore.CurrentCities.Cities.Where(x => x.Id == id).FirstOrDefault();

if (item == null)

{

return NotFound();

}

item.Name = VM.Name;

item.Description = VM.Description;

var Data = CityDataStore.CurrentCities.Cities;

return Ok(Data);

}

catch (Exception)

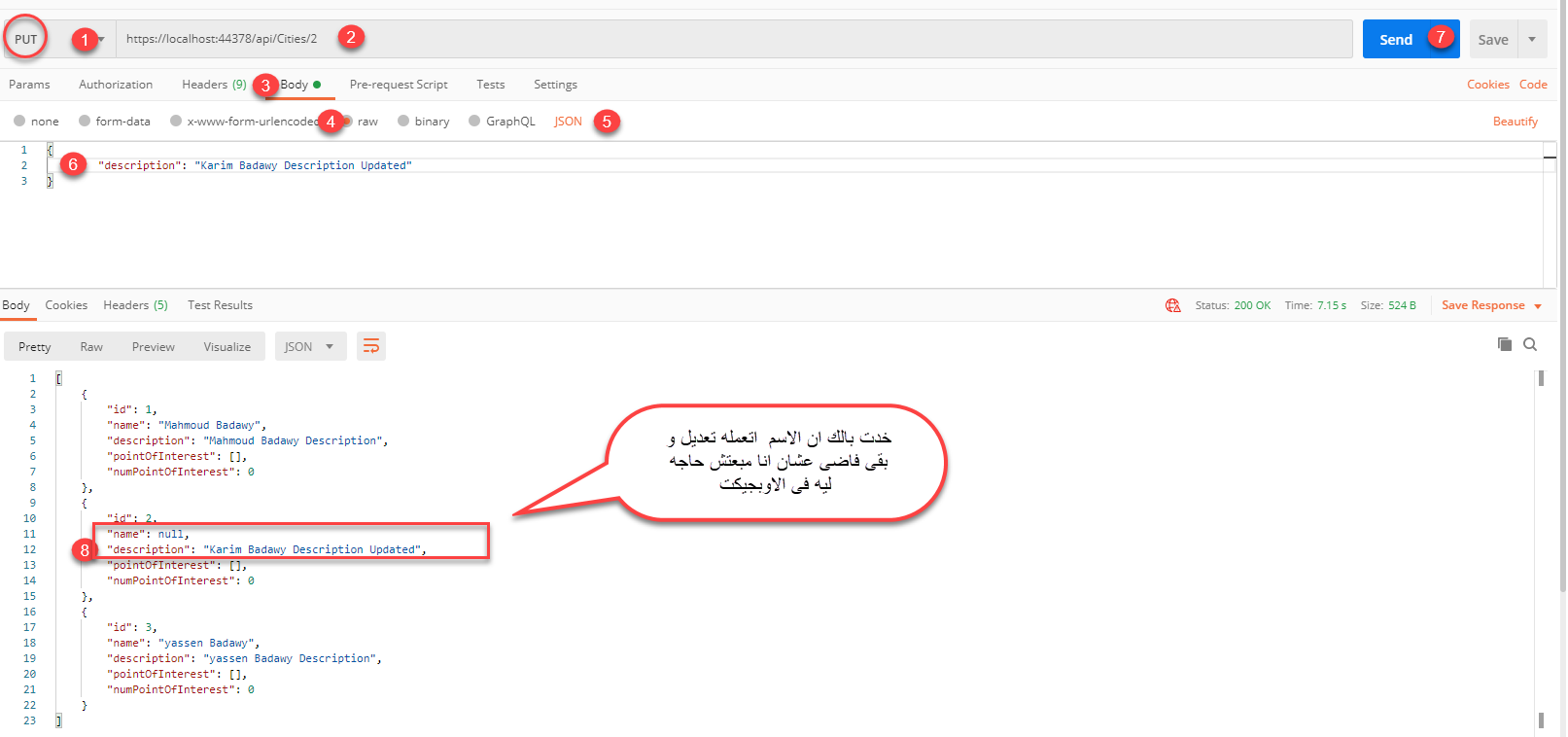
{

return BadRequest();

throw;

}

}



بينما فى ال Patch بقه هيعدل فى الديسكريبشن و ملوش دعوه بالباقى   
// Patch: api/Cities/5

[HttpPatch("{id}")]

public IActionResult Patch(int id, JsonPatchDocument<CityDto> VM)

{

try

{

CityDto item = CityDataStore.CurrentCities.Cities.Where(x => x.Id == id).FirstOrDefault();

if (item == null)

{

return NotFound();

}

CityDto PatchObj = new CityDto()

{

Name = item.Name,

Description = item.Description

};

VM.ApplyTo(PatchObj, ModelState);

if (!ModelState.IsValid)

{

return BadRequest(ModelState);

}

item.Name = PatchObj.Name;

item.Description = PatchObj.Description;

var Data = CityDataStore.CurrentCities.Cities;

return Ok(Data);

}

catch (Exception ex)

{

return BadRequest();

throw;

}

}

اول تغيير فى الكود ان الحاجه الى ببعتها بتبقى لست من نوع JsonPatchDocument للداتا بتعتى الى من نوع CityDto  
تانى حاجه انا بخزن اول حاجه الاوبجيكت الى هعدل عليه فى ال Item   
ثم بعمل PatchObject ذى متقول كده TempObject اخزن فيه ال Item بعد كده بعمل بقوله اعملى ApplyTp من التعديل الى انا جيبهولك للتيمب أو الباتشيد اوبجيكت بحيث التعديلات تتنقل من الحاجه المبعوته من الكلاينت لل PatchObj و بكده حصل ريبليس خلاص طيب بما ان التغيير تم على ال PatchObj فانا دلوقتى اقدر انقله بقه لل Item تانى و بكده شكرا التعديل تم فقط فى الحاجه الى انا حددتها   
  
طيب ارسال الاوبجيكت هنا من نوع JsonPatchDocument<CityDto> و بالتالى انا هبعته بشكل خاص بيبقى عباره عن Array اوف Object و الاوبجيكت فيه نوع ال Operation و البروبرتى الى هيخير فيها باكى Path و القيمه بتعتها بقه فى ال value   
بالشكل ده ذى ما هتشوف فى بوست مان   
و خد بالك ان ال op ممكن تبقى replace او remove و ممكن تغيير كذا حاجه مع بعض بانك تبعت كذا اوبجيكت فى الاراى الى بتبعته   
جرب ده كده مره

[{

        "op":"replace",

        "path":"/description",

        "value": "Karim Badawy Description Updated"

}]

و بعد كده ده

[{

        "op":"replace",

        "path":"/description",

        "value": "Karim Badawy Description Updated" } ,

{

        "op":"replace",

        "path":"/name",

        "value": "Karim Badawy name Updated"

}]

و كذلك جرب دى

[{"op":"replace",

        "path":"/description",

        "value": "Karim Badawy Description Updated"

}

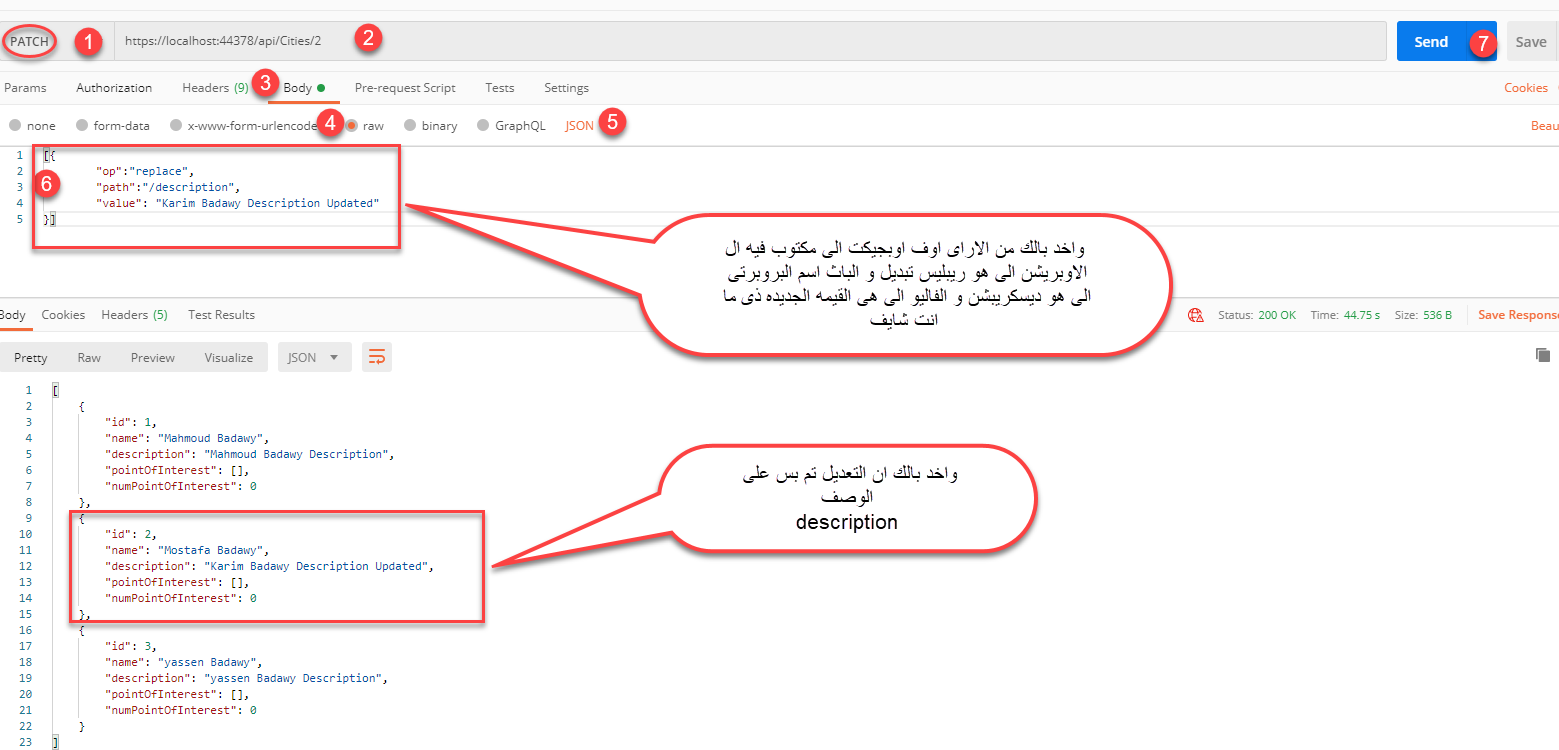
,

{

        "op":"remove",

        "path":"/name"

}]



طب لو عاوزه ميعملش ريموف لحاجه المفروض اعمل حاجتين اقوله فى ال dataAnottaction ان الحاجه دى Required و اقوله فى الكود السطرين دول

if (!TryValidateModel(PatchObj))

{

return BadRequest(ModelState);

}

MB

################################################################

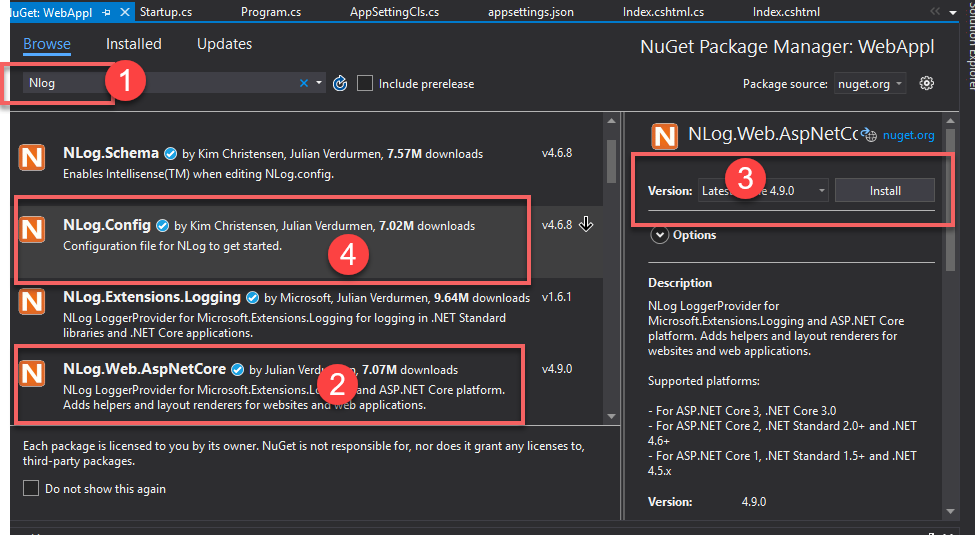
Log File

---------------------------------------------------------------

بص يا سطى انت فى فايل كده بتحفظ فيه كل الايرورز تمام ده بقه ليه باكيدج حلو جدا دلوقتى اسمه NLog.Web.AspNetCore

هتروح تسطبه اذاى بقه ؟؟

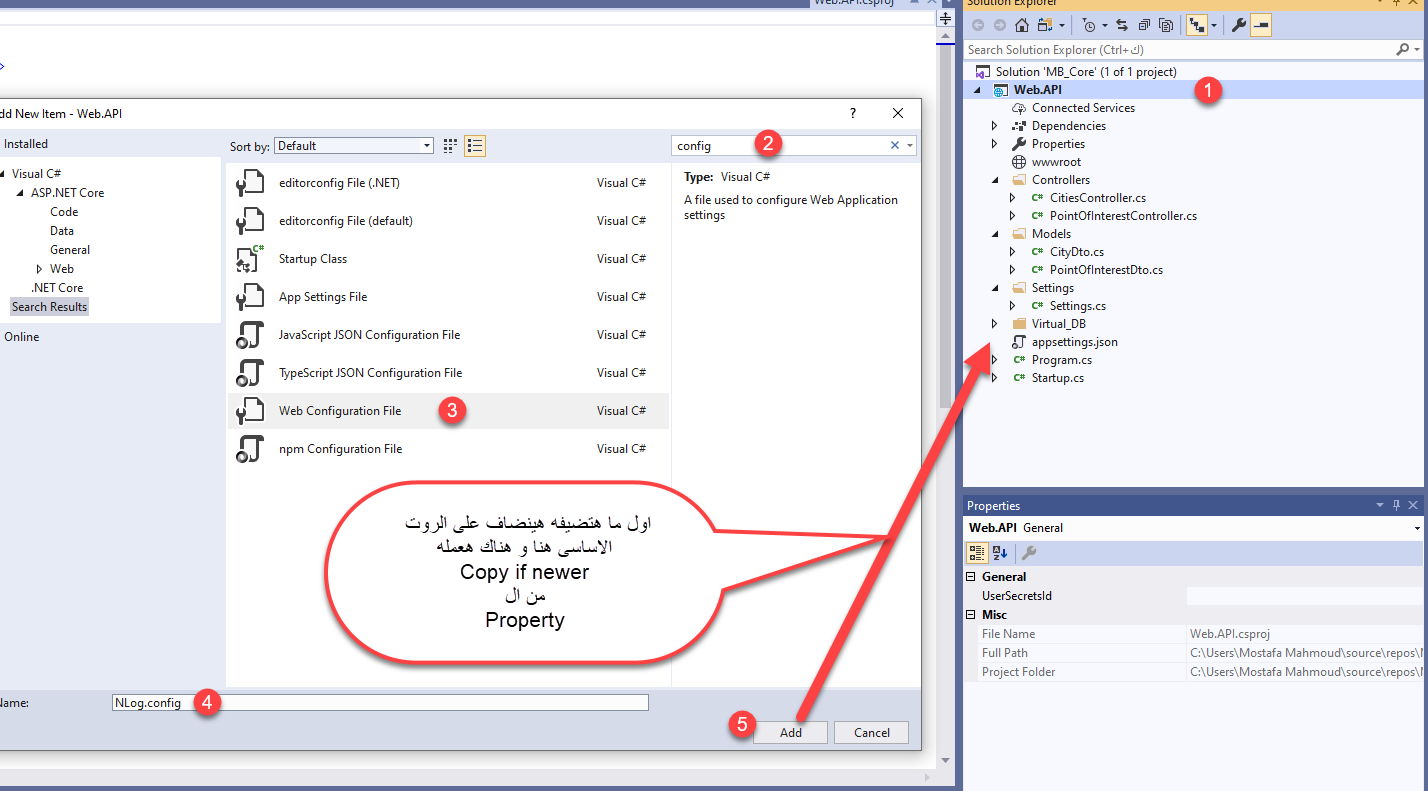
هتروح على ال Dependencies او ال References الى فى المشروع بتاع ال Api و تقوله add By Nuget و هتخش على Browse و تكتب الاسم هتظهرلك سطبها ذى ما انت شايف هنا كده



بعد كده المفروض بتكريت ال NLog.Config بطريقه من الاتنين

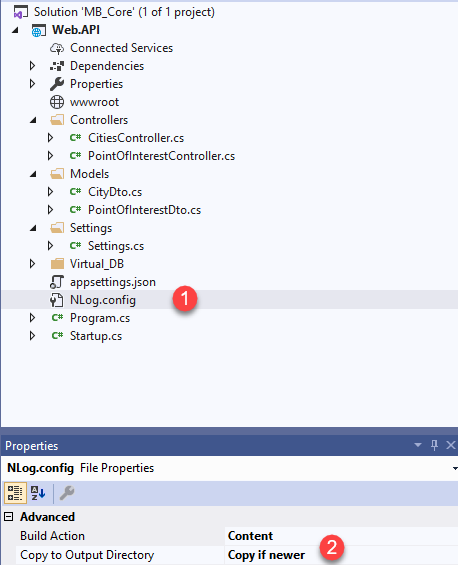
الطريقه الاولى انى اسطبها من الباكيدجات

الطريقه التانيه انى اعملهما Manulay اروح اعمل فايل بالاسم ده على الروت و انا بفضل بصراحه اكريته مانوال عشان لو عملته اوتوماتيك التحكم فيه بيبقى بيض و اخش اعدل فيه بقه براحتى بعد كده



Advised to: download manually: <https://raw.githubusercontent.com/NLog/NLog/dev/src/NuGet/NLog.Config/content/NLog.config>

- set "Copy To Output Directory" to "Copy if newer"



NLog.Config

نروح بقه نملى الكونفيجريشن ذى ما انت شايف كده

<?xml version="1.0" encoding="utf-8" ?>

<nlog xmlns="http://www.nlog-project.org/schemas/NLog.xsd"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.nlog-project.org/schemas/NLog.xsd NLog.xsd"

autoReload="true"

throwExceptions="false"

internalLogLevel="Off" internalLogFile="c:\temp\nlog-internal.log">

<!-- optional, add some variables

https://github.com/nlog/NLog/wiki/Configuration-file#variables

-->

<variable name="myvar" value="myvalue"/>

<!--

See https://github.com/nlog/nlog/wiki/Configuration-file

for information on customizing logging rules and outputs.

-->

<targets>

<!--

add your targets here

See https://github.com/nlog/NLog/wiki/Targets for possible targets.

See https://github.com/nlog/NLog/wiki/Layout-Renderers for the possible layout renderers.

-->

<!--

Write events to a file with the date in the filename.

-->

<target xsi:type="File" name="f" fileName="${basedir}/logs/${shortdate}.log"

layout="${longdate} ${uppercase:${level}} ${message}" />

</targets>

<rules>

<!-- add your logging rules here -->

<!--

Write all events with minimal level of Debug (So Debug, Info, Warn, Error and Fatal, but not Trace) to "f"

-->

<logger name="\*" minlevel="Debug" writeTo="f" />

</rules>

</nlog>

و بما انى هشتغل على Core 2 فهخش على الموقع اجيب الكونفجريشن منه من هنا

<https://github.com/NLog/NLog/wiki/Getting-started-with-ASP.NET-Core-2>

و لو هشتغل على Core 3 فهخش على الموقع اجيب الكونفجريشن منه من هنا

<https://github.com/NLog/NLog/wiki/Getting-started-with-ASP.NET-Core-3>

هخش بقه و اتبع التعليمات بالحرف :D و فى الغالب انت هتشغل الحاجات الى جوه الفايل الاوتوماتيك يعنى هتشيل من عليها ال comments بس باقى الشغل هتعمله ذى التعليمات بالظبط يا اسطى :D

و متنساش تبص هنا برضوا عشان تعرف تعدل على الفايل براحتك

<https://github.com/NLog/NLog/wiki/Configuration-file>

طيب دلوقتى انا سطبت كل حاجه مش ناقص غير انى استخدم ال NLOG بقه   
بص يا سيدى اول حاجه عشان ال NLOG يشتغل على البروجيكت فهروح على فايل ال Programe.cs و اضيفه هناك على ال startup فانكشن كده

namespace Web.API

{

public class Program

{

public static void Main(string[] args)

{

var logger = NLogBuilder.ConfigureNLog("NLog.config").GetCurrentClassLogger();

try

{

logger.Info("Intiallize App");

}

catch (Exception ex)

{

logger.Error("App Stopped Because of Exception");

throw;

}

finally

{

//السطر ده هيوقفلك شغل اللوجر فى البروجيكت كله لانه اول حاجه بيعدى عليها و هنا بقوله اقفلى اللوجر

//فبلاش تستخدمه ملوش لازمه

//NLog.LogManager.Shutdown();

}

CreateWebHostBuilder(args).Build().Run();

}

public static IWebHostBuilder CreateWebHostBuilder(string[] args) =>

WebHost.CreateDefaultBuilder(args)

.UseStartup<Startup>().UseNLog();

}

}

كده اللوجر اشتغل على البروجيكت كله و هيطلع الايرورز فى الفايل لوحده بس بفرض بقه انك عاوز تعمل رسايل مخصصه ليك تعمل ايه ؟؟  
بما ان NLog دى Built in Service فانا كده اقدر استخدمها فى اى كلاس عن طريق انى اعملها Dependency Injection فى الكلاس و منه اقدر استخدمها فيه بكل سهوله و العب فيها براحى و تعالى نستخدمها فى الكنترولار بتاع ال CitiesController كده

namespace Web.API.Controllers

{

[Route("api/Cities")]

[ApiController]

public class CitiesController : ControllerBase

{

// Dependency Injection By IOC

private readonly ILogger<CitiesController> \_logger;

public CitiesController(ILogger<CitiesController> logger)

{

\_logger = logger;

}

// GET: api/Cities

[HttpGet]

public IActionResult Get()

{

try

{

Exception x = new Exception();

throw x;

List<CityDto> Data = CityDataStore.CurrentCities.Cities.ToList();

if (Data == null)

{

return NoContent();

}

return Ok(Data);

}

catch (Exception)

{

\_logger.LogInformation("MW Finshed");

\_logger.LogError("Error ya lahwy");

return BadRequest();

}

}

}

}

MB

################################################################

**Dependency Injection By IOC**

---------------------------------------------------------------

الدبندينسى انجيكشن ده باترن اتعمل عشان استخدم اى كلاس بسهوله و مش كل شويه اعملها اوبجيكت و كل اما اغير فى الكونستراكتور بتاع الكلاس اجرى اغير فى كل الاوبجيكتس بتعتها فى المشروع كله و بالتالى هى حاجه مفيده جدا جدا جدا

و طريقه استخدامها انى اولا بعمل بوربرتى من الحاجه الى انا هعملها انجيكت و من ثم اعمل الانجيكشن فى الكونستراكتور و بعد كده اخد القيمه الى فى الانجيكشن احطها فى البروبرتى و بعد كده استخدمها فى الكلاس براحتى   
غالبا بيبقى ال DI لكلاس Service عشان يوفر عليا شغل كتير و خلى بالك ال DI فى ال Core جاهز بيلت ان يعنى غير زمان كنت بتبقى مضطر تعمله بنفسك و تسطبله باكيدجات و حوارات بقه

انا عمتا استخدمت ال DI اول استخدام مع سيرفيس ال Logger عن طريق الانترفيس بتعها الى هو ILogger هنا عملت انجيكشن لحاجه من نوع لسته لوجر من الكنترول كل سيرفيس بقه ليها طريقه كتابه بس بشكل عام انت بتعمل الانجيكشن فى الكنستراكتور ذى ما انت شايف كده و هنتعرض لكذا حاجه هتشوفها برضوا لا تقلق

// Dependency Injection By IOC

private readonly ILogger<CitiesController> \_logger;

public CitiesController(ILogger<CitiesController> logger)

{

\_logger = logger ?? throw new ArgumentNullException(nameof(logger));

}

و بعد كده بتستخدم ال DI فاريبول فى اى مكان فى الكلاس بقه خلاص

[HttpGet]

public IActionResult Get()

{

try

{

Exception x = new Exception();

throw x;

List<CityDto> Data = CityDataStore.CurrentCities.Cities.ToList();

if (Data == null)

{

return NoContent();

}

return Ok(Data);

}

catch (Exception)

{

\_logger.LogInformation("MW Finshed");

\_logger.LogError("Error ya lahwy");

return BadRequest();

}

}

تعالى بقه نعمل سيرفيس بنفسنا و نجرب عليها ال DI

?? throw new ArgumentNullException(nameof(logger)); \_logger = logger

بس الاول خلى بالك ان الجزء ده فى الانجيكشن مهم عشان لو فى مره الحاجه مشتغلتش يعرفك المشكله فين بسرعه

MB

################################################################

**How to make Services and Use it**

---------------------------------------------------------------

أول حاجه هعمل Folder اسمه Services احط فيه اى سيرفيس هعملها و السيرفيس هنا عشان نبقى كاتبين كود نضيف بنقسمها لفايلين فايل الكلاس العادى و فايل الانترفيس بتاعه   
LocalGetCities\_service و ده فايل الكلاس بتاعت السيرفيس الاساسى الى هعمل فيه الامبليمنتيشن و اخلصه هنا و بعد ما اخلص امبليمنتيشن ذى ما انت شايف كده

namespace Web.API.Services

{

public class LocalGetCities\_service

{

public List<CityDto> GetCities()

{

try

{

List<CityDto> Data = CityDataStore.CurrentCities.Cities.ToList();

if (Data != null)

{

return Data;

}

else

{

return null;

}

}

catch (Exception)

{

throw;

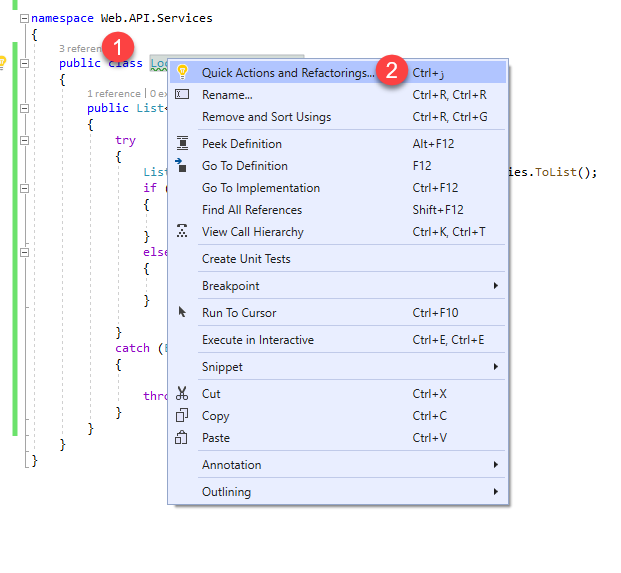
}

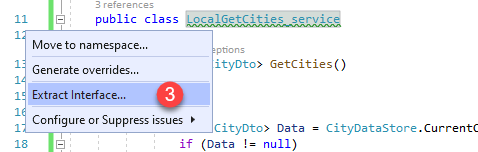
}

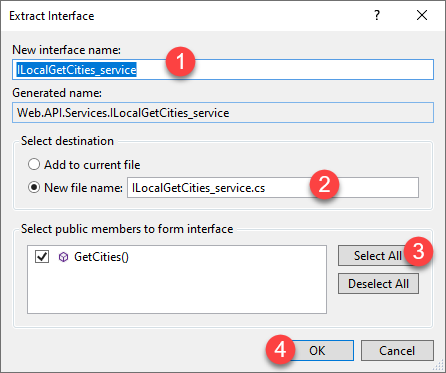
}

}

هروح عامل ال Interface منه بالطريقه دى بروح جاى على الكلاس معلم علي اسمه فوق كده و كليك يمين و ادوس Quick Action و بعد كده ادوس على Extract Interface





و هتختار اسم الانترفيس و طبعا هتقوله يعملك الديفينيشن بتاعت كل الفانكشنز بتعتك ذى كده   


بعد كده هتخلى الكلاس الاصليه بتورث من ال interface

namespace Web.API.Services

{

public class LocalGetCities\_service : ILocalGetCities\_service

{

public List<CityDto> GetCities()

{

try

{

List<CityDto> Data = CityDataStore.CurrentCities.Cities.ToList();

if (Data != null)

{

return Data;

}

else

{

return null;

}

}

catch (Exception)

{

throw;

}

}

}

}

كده انت عملت السيرفيس المفروض تستخدمها بقه فى الكنترولار بتاعك فهتسخدمها كده

namespace Web.API.Controllers

{

[Route("api/Cities")]

[ApiController]

public class CitiesController : ControllerBase

{

private readonly ILocalGetCities\_service \_LocalGetCities;

public CitiesController( ILocalGetCities\_service LocalGetCities)

{

\_LocalGetCities = LocalGetCities;

}

// GET: api/Cities

[HttpGet]

public IActionResult Get()

{

try

{

List<CityDto> Data = \_LocalGetCities.GetCities();

if (Data != null)

{

return Ok(Data);

}

else

{

return BadRequest();

}

}

catch (Exception)

{

return BadRequest();

}

}

}

}

طيب دلوقتى انت استخدمت ال services فى الكنترولار بال Dependency Injection و لاحظ انى استخدمت هنا الانترفيس interface مش الكلاس الاساسيه عشان يبقى معمول انكابسوليشن للبيانات بشكل رهيب بس لو رنيت على كده مش هيرن و هيبوظ

طيب المفروض ناقص كده حاجه واحده انك تروح تسجل ال Services دى فى ال Startup.cs فى الكونيفجر سيرفيس فانكشن و دى هتتسجل هناك كده

public void ConfigureServices(IServiceCollection services)

{

services.AddMvc().AddMvcOptions(X => {

X.OutputFormatters.Add(new XmlDataContractSerializerOutputFormatter());

});

//حط السيرفيس بتعتك هنا

services.AddTransient<ILocalGetCities\_service, LocalGetCities\_service>();

}

هنا و انا بعرف السيرفر بقوله addTransient على السيرفيسيس من نوع الانترفيس ILocalGetCities\_service

و كل اما تشوف ال الانترفيس ده روح استخدم الكلاس بتاعه بقه الى هو ده LocalGetCities\_service

بكده جرب بقه يا سيدى السيرفيس على كده هتلاقيه اشتغلت الله ينور

جرب بقه و شوف الحلاوه

**ملحوظه انت تقدر تعمل اتنين سيرفيس بتعمل انهيرتانس من interface واحده لو فيها نفس الامبليمينتيشن بس واحده هستخدمها فى debug و التانيه هستخدمها فى ال production**

فمثلا هعمل سيرفيس تانيه بتعمل انهيريت من ال ILocalGetCities\_service و هسميها ProductionGetCities\_service و الكود الى فيها هخليه هو هواه ذى بتاعت الديباج بس هكتب جوه السيرفيس لوجر يعرفنى انه دخل الفايل بتاع البروداكشن و خلاص

namespace Web.API.Services

{

public class PeoductionGetCities\_service : ILocalGetCities\_service

{

private ILogger<PeoductionGetCities\_service> \_logger;

public PeoductionGetCities\_service(ILogger<PeoductionGetCities\_service> logger)

{

\_logger = logger ?? throw new ArgumentNullException(nameof(logger));

}

public List<CityDto> GetCities()

{

try

{

\_logger.LogInformation("Hi From Production");

List<CityDto> Data = CityDataStore.CurrentCities.Cities.ToList();

if (Data != null)

{

return Data;

}

else

{

return null;

}

}

catch (Exception)

{

throw;

}

}

}

}

بس كده هروح و انا بسجل السيرفيس هعمل حاجه اسمها if Debug

#if DEBUG

services.AddTransient<ILocalGetCities\_service, LocalGetCities\_service>();

#else

services.AddTransient<ILocalGetCities\_service, PeoductionGetCities\_service>();

#endif

شغلت المشروع فى debug mode هتلاحظ انه معملش اى لوج اصلا يبقى كده هو دخل الفايل العادى

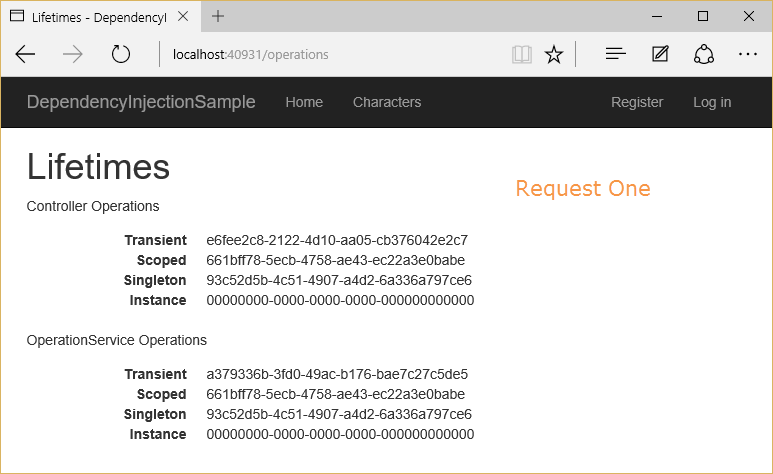
شغلت المشروع فى release Mode هتلاحظ انه كتب اللوج الى انت عامله فى الفايل بتاع البروداكشن يا دهكله و ده معناه انه شغل السيرفيس الخاصه بالبروداكشن و ساب بتاعت اللواكل و هتلاقى اللوج بتاعه موجود فى ال release folder عشان متتلعبكش بس

ملاحظات هامه جدا   
ايه الفرق بين السرفيس الى بعملها addTransient و addScoped و addSinglton

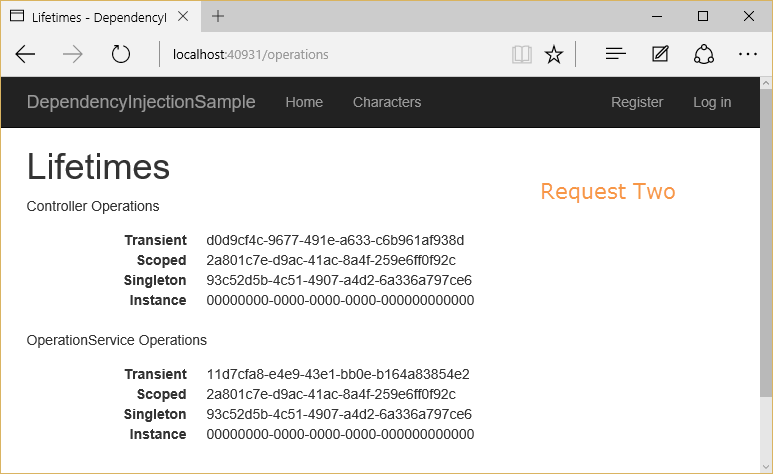
**addTransient** : السيرفيس بتعتها فى كل مره بيتعمل فيها ريكويست بتكريت Object جديده خالص عن الريكويست الى قابله حتى لو الريكويست بتتنادى كذا مره و انت بتفتح الصفحه فهيبقى على الرغم ان الريكويست هى ريكويست واحده الى حصلت الا ان عمل كرييت للاوبجيكت كل مره مشى فيها فى الريكويست و هيحصل انه هيعمل اتنين اوبجيكت مختلفين تماما

**addScoped** : هنا بقه السيرفيس بتعتها طول ما هو بينادى نفس الريكويست حتى لو ندها مرتين ال Object بيتغيرش و بيفضل هو نفسه بس اول ما يغيير الريكويست و يبعت ريكويست تانى حتى لو هو هو الريكويست بس بعت ب http جديد فساعتها هيعتبره ريكويست مختلف و بالتالى ال Object بيتغير ساعتها

**addSinglton** : ده بقه الاوبجيكت بيتغيرش خالص حتى لو غيرت الريكويست و بعت ريكويست جديد تماما هيفضل هو هواه برضوا معفن يعنى ههههههه

الصوره دى بتبين و دى عباره عن صوره بتوريك الفرق لما حصل ان ريكويست اتعملت فى صفحه بس المفروض انها هتحط نتايج فى مكانين فكان نتيجه اول ريكويست اختلاف اوبجيكت ال Transient فى المكانين لنفس السيرفيس   
تساوى اوبجيكت ال addSctoped فى المكانين لنفس السيرفيس و كذلك تساوى الاوبجيكت فى ال addSinglton فى المكانين لنفس السيرفيس   


اما فى الريكويست التانى هتلاقى ان ال اختلاف اوبجيكت ال Transient فى المكانين لنفس السيرفيس و كذلك اختلاف اوبجيكت ال addSctoped فى المكانين لنفس السيرفيس بينما هيفضل المعفن addSinglton مبيتغيرش برضوا فى الريكويست التانى لنفس السيرفيس



# Which one to use

**Transient**

* since they are created every time they will use **more memory** & Resources and can have the negative impact on performance
* use this for the **lightweight** service with little or **no state**.

**Scoped**

* better option when you want to maintain state within a request.

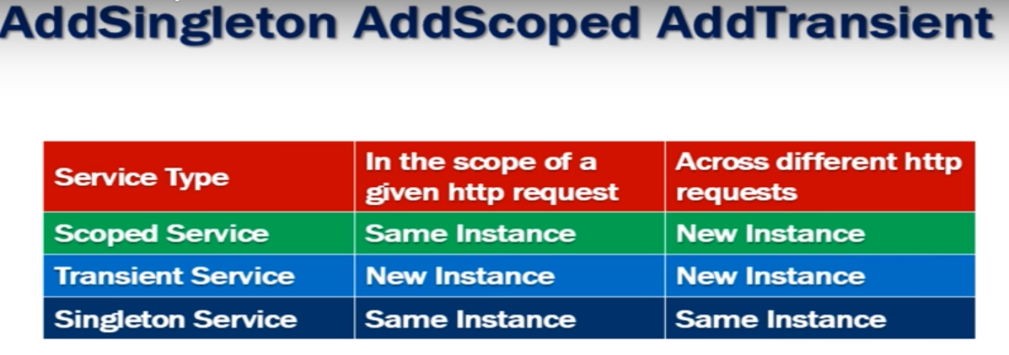
**Singleton**

* memory leaks in these services will build up over time.
* also memory efficient as they are created once reused everywhere.

Use Singletons where you need to maintain application wide state. Application configuration or parameters, Logging Service, caching of data is some of the examples where you can use singletons.

## **Injecting service with different lifetimes into another**

1. **Never inject Scoped & Transient services into Singleton service.** ( This effectively converts the transient or scoped service into the singleton.)
2. **Never inject Transient services into scoped service** ( This converts the transient service into the scoped. )



خد بقه اللينك ده هتفهم منه فشخ الجزئيه دى

<https://stackoverflow.com/questions/38138100/addtransient-addscoped-and-addsingleton-services-differences>

MB

################################################################

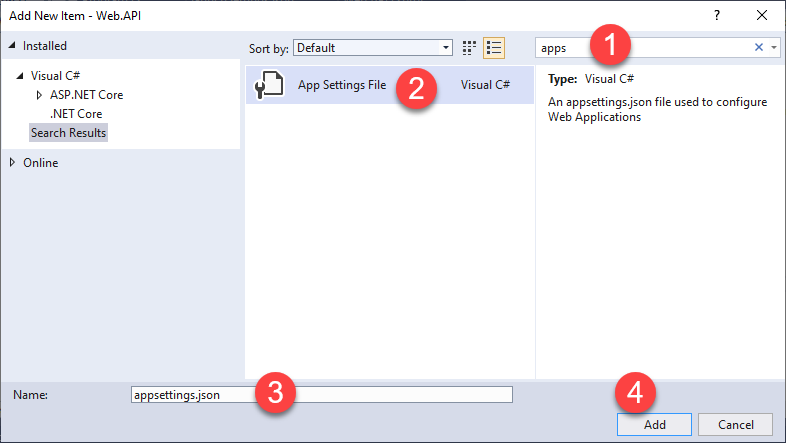
Working with Configuration Files

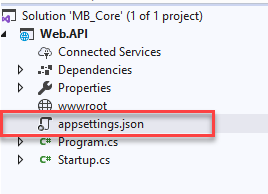
---------------------------------------------------------------

اذاى اعمل كونفيجريشن فايل للكويرى سترينج و احط فيها حاجات كتير بمزاجى و اذاى اقرا الى جواها كله   
اولا هعمل ملف اسمه AppSetting.json هو ده الى بعمل فيه الكونفيجريشن بتعتى كلها

**Appsettings** ده بقه ذى ما تقول كده ده ال wep config بشكله الجديد هههههههههه يعنى تحط فيه connection string و المشروع كله هيبقى شايفه و ده عشان يوفر عليا انى اعد اكتب الكونيكشن سترينج مليون مره فى الموقع و لو حبيت اغيره اغيره فى المكان ده بس و ابقى مش مضطر اغيره فى مليون مكان و ده بقه تحط فيه اى حاجه مهمه سواء خاصه بالابليكيشن او السيرفر الى هيترفع عليه المشروع و بيقى كتابته Json بالشكل ده

و دى بحطها تحت البروجيكت على طول يعنى فى اول ليفل جوه فولدر المشروه جنب Startup.cs , program.cs





{

"Logging": {

"LogLevel": {

"Default": "Information",

"Microsoft": "Warning",

"Microsoft.Hosting.Lifetime": "Information"

}

},

"AllowedHosts": "\*",

"ConnectionString": "My Connection String Info",

"ProjectName" : {

"DefaultName": "Core Project",

"RealName" : "Badawy Project"

}

}

طيب انا كتبت الكلام ده فى ال AppSettings.json عاوز اقراه بقه من اى مكان بقه اعمل ايه ؟؟

بص يا سيدى انا هقراه فى ال Startup.cs فتعالى اوريك هيتقرأ اذاى

namespace Web.API

{

public class Startup

{

private IConfiguration Configuration { get; }

public string ConnString { set; get; }

public Startup(IConfiguration configuration)

{

Configuration = configuration;

}

public void ConfigureServices(IServiceCollection services)

{}

public void Configure(IApplicationBuilder app, IHostingEnvironment env)

{

if (env.IsDevelopment())

{

app.UseDeveloperExceptionPage();

}

app.Run(async (context) =>

{

var ProjectName = Configuration["ProjectName:RealName"];

ConnString = Configuration["ConnectionString"];

await context.Response.WriteAsync(ProjectName + " -- " + ConnString);

});

}

}

}

ذى ما انت شايق يا كبير عشان تقرأ حاجه فى ال AppSettings بتروح رايح على الكلاس الى انت هتقرا فيه من ال AppSettings و عامل Dependency Injection من نوع IConfigration و كده يا كبير انت بقى سهل تقرأ من ال AppSettings اى حاجه فيها و تقدر طبعا تسجلها فى اى فاريبول او بروبارتى او اى فيو داتا و تعرضها فى اى حته او تستخدمها باى شكل بقه براحتك

كده اول ما هتعمل Run هيروح يكتب فى البراوزار اسم المشروع و الكونيكشن سترينج بتاعه

ملحوظه و انت بتقرأ اى حاجه مش مباشره من ال AppSetting هتعمل بين كل parent , Child نقطتين فوق بعض مش دوت ذى ما انت شايف فى المثال ده

var ProjectName = Configuration["ProjectName**:**RealName"];

MB

################################################################

Scoping Configuration to Environments

---------------------------------------------------------------

ده يعنى اذاى اعمل كةنفيجريشن لكذا Environment يعنى يبقى مثلا كونيجريشن خاص باللوكال و واحد تانى للبروداكشن   
بص يا سيدى كل الى هتعمله انك هتعمل فايل جديد على المشروع هيبقى اسمه كده   
appsettings.production.json  
و تحط فيه الكونفيجريشن بنفس الهيراركى بتاعت   
appsettings.json

يعنى يبقى الاتنين نسخه من بعض بس التغيير فى القيم بس مش اكتر ذى كده

appsettings.json

{

"ConnectionString": {

"DefaultConnStr": "My Connection String Info"

},

"ProjectName": {

"DefaultName": "Core Project",

"RealName": "Badawy Project"

}

}

اذا الفايل بتاع البروداكشن هيبقى كده

appsettings.production.json

{

"ConnectionString": {

"DefaultConnStr": "production Connection String Info"

},

"ProjectName": {

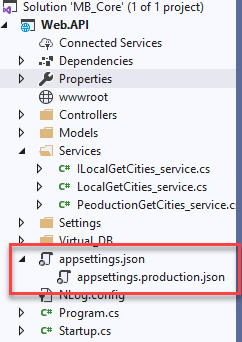
"DefaultName": "Core production Project",

"RealName": "Badawy production Project"

}

}

و هيبقى شكلهم فى البروجيكت ذى الصوره دى كده



**و لو عاوز تجرب بقه البروداكشن من غير ما ترفعه كل الى عليك هتدوس على properties المشروع و تخش على Debug و من ثم تقوم بتغيير قيمه ال ASPNETCORE\_ENVIRONMENT من ال Development الى production**

**و على فكره ممكن تعمل فايل للديفولب مينت فقط خاص بيه و تسميه كده**

**appsettings.Development.json**

**و طول ما انت شغال فى الديفولوب مينت هو ده الى هيرن يا دهلكه**

MB

################################################################

Extention Methods

---------------------------------------------------------------

هى نوع من الفانكشنز بضيف ليها الكلاس عشان تشتغل عليه مش العكس يعنى مش بضيفها جوه الكلاس عشان تشتغل فيه ههههههههههه يعنى دى يا سيدى نوع جديد من ال Methods بيسهل عليا كتابه الفانكشن الى هنفذها على كلاس معين بدون ما اكتب الفانكشن دى جوه الكلاس الى هتشتغل عليه

يعنى بدل ما اضطر انى اعمل فانكشن جوه كلاس ما و عشان استخدمها اضطر اعمل منه اوبجيكت و اناديها و الكلام الفارغ ده فقالك و على ايه منا استخدم ال Extention Method و اضيف الكلاس جواها كبراميتار ليها و هى تتعامل معاه و ترجع النتيجه منه و خلاص

اغلب استخدامها بيبقى فى انى اعمل فانكشن مثلا بتجيب ال MaxId من الجدول الى انا عامل كلاس ال Entity بأسمه يعنى غالبا بنستخدمها فى الفانكشنز الى هستخدمها كتير فى ال Classes

**مثال** دلوقتى انا عاوز اديك string فهو ياخد ال String ده و يحوله ل integer

طيب انا دلوقتى عشان اعمل الكلام ده هيبقى مكلف اوى لو بالطريقه العاديه لانى هبقى مضطر اروح ادخل جوه كلاس ال string و اعمل فيه الفانكشن دى و بعد كده اعمل اوبجيكت من نوع string و بعد كده اروح انادى للفانكشن الجديده الى انا عاملها عشان تحول الاسترينج لانتيجر يعنى حاجه خره من الاخر

الطريقه التقليديه اهه و عفانتها

namespace MyConsoleApp

{

class Program

{

static void Main(string[] args)

{

cls MyObj = new cls();

int Num = MyObj.ConvertStrToInt("123456");

Console.WriteLine(Num);

}

}

public class cls

{

public int ConvertStrToInt(string MyStr)

{

return int.Parse(MyStr);

}

}

}

طريقه الاكستينشان بقه

namespace MyConsoleApp

{

class Program

{

static void Main(string[] args)

{

string MyStr = "123456";

int Num = **MyStr.ConvertStrToInt();**

Console.WriteLine(Num);

}

}

public **static** class ExtentionsMethodsForString

{

public **static** int ConvertStrToInt(**this** string MyStr)

{

return int.Parse(MyStr);

}

}

}

ذى ما انت شايف فوق عمالنا كلاس اسمها ExtentionsMethodsForString دى فقط لمجرد انى بحفظ الاكستنشان فى مكان منظم و خلاص :D لانى فى الغالب مش هناديها بيه حتى و طبعا الكلاس دى لازم تبقى

**كلاس** **Static** بعد كده بعمل **الفانكشن** و برضوا لازم تبقى **Static** هى كمان و بعد كده بدى الفانكشن دى Parameter بيبقى عباره عن this و ده معناه ده هو الاوبجيكت الى من نوع الداتا الى هيتعامل معاها و هى بيحددها جنبه على طول و هنا هتبقى String و طبعا بيديله اسم البراميتار الى هيشتغل عليه و ده يعتبر الاسترينج الى هو هيبقى الاكستينشن فانكشن بتاعه البراميتار ده مش بيتم استخدامه فى ال call و لكن لو حابب بقه تضيف له براميتار هتستخدمه فى ال Call بتضيف كمان واحد بقه يا باشا من غير ال this

و ادى مثال اهه يا هم هياخد الرقم و يقسمه على الرقم الى انت هتديهوله من البراميتار ياباشا و ليكن انا هنا هثبته ب 1000

using System;

namespace MyConsoleApp

{

class Program

{

static void Main(string[] args)

{

string MyStr = "123456";

float Num = **MyStr.ConvertStrToInt(1000);**

Console.WriteLine(Num);

}

}

public static class ExtentionsMethodsForString

{

public static float ConvertStrToInt(**this string MyStr , int DivNum**)

{

return (float.Parse(MyStr) / DivNum);

}

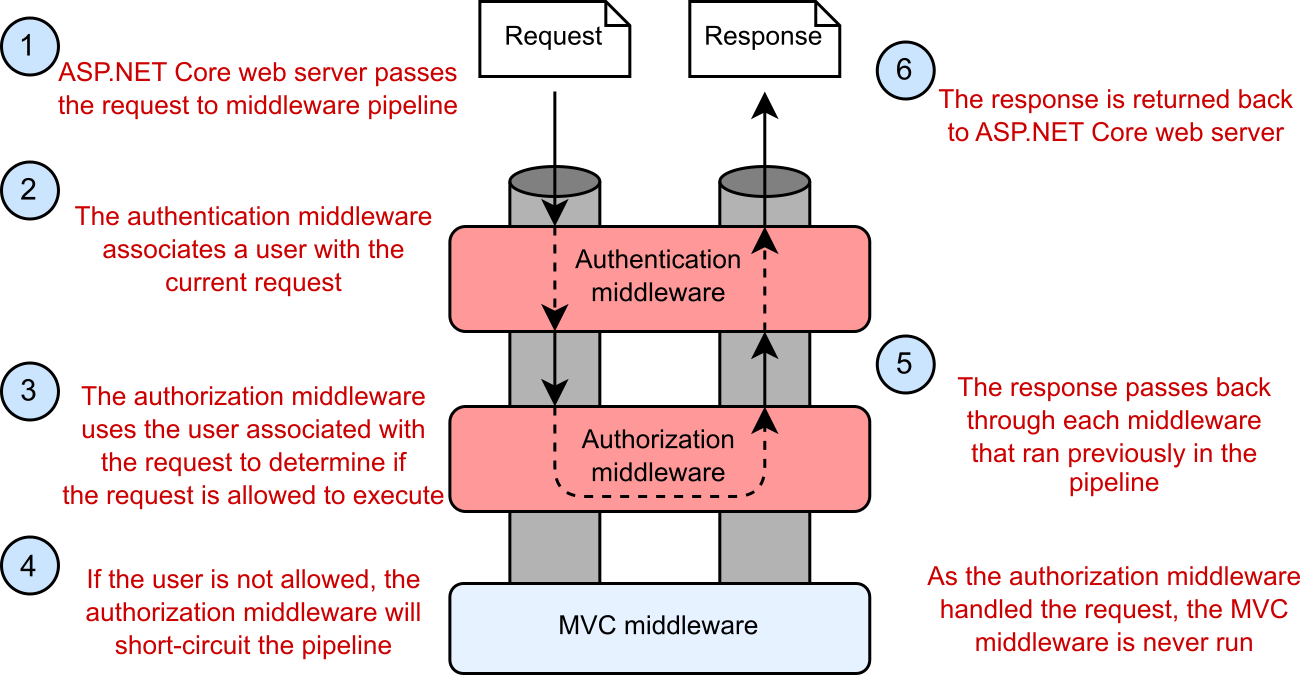
}

}

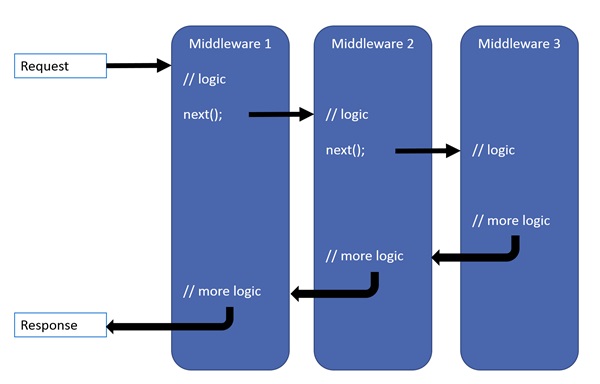
MB

##########################################################################

Middlewares

------------------------------

الميديل وير ده وسيط لازم يمر عليه اى ريكويست بتحصل بين ال Client و ال Server و فيها بقدر اعمل اى بروسيس اثناء الفتره بين الريكويست ومن الكلاينت للسيرفر لحد ما السيرفر يرد على الكلاينت بريسبونس بص هى من الاخر عامله ذى ال interceptor فى Angular بالظبط كان الميدل ويرز الى موجوده فى الموقع دى هى مواسير المايه الى المايه بتمشى فيها   
فعشان كده اى ريكويست بتحصل لازم تعدى على كل ال Middlewares و من اهم الميدل ويرز فى الموقع الميديل وير بتاع ال Auth لان فى كل مره ال Client بيعمل ريكويست الميديل وير دى بتشوف الريكويست دى و ترد عليه اشطه انت عديت و نجحت او ترد برضوا عليه و تقوله ياباشا انت معدتش و منجحتش



**بالنسبه للكود فهو اى ميدل وير هعملها بحط الريجيستريشن بتعها فى ال Startup.cs** فى الفانكشنن دى **Configure وده مثال من على النت عشان ننجز**

namespace TodoApi

{

public class Startup

{

public Startup(IConfiguration configuration)

{

Configuration = configuration;

}

public IConfiguration Configuration { get; }

public void ConfigureServices(IServiceCollection services)

{

services.AddDbContext<TodoContext>(opt => opt.UseInMemoryDatabase("TodoList"));

services.AddControllers();

}

public void Configure(IApplicationBuilder app, IWebHostEnvironment env)

{

if (env.IsDevelopment())

{

app.UseDeveloperExceptionPage();

}

app.UseHttpsRedirection();

app.UseRouting();

app.UseAuthorization();

app.UseEndpoints(endpoints =>

{

endpoints.MapControllers();

});

}

}

}

**خد بالك ان الترتيب مهم جدا فى الميدل وير يعنى عشان ممكن يبقى في واحد بيعتمد على التانى فلازم يبقوا مترتبين صح يعنى مثلا عندى ميدل وير خاصه بالدخول على الموقع بالايميل و الباسورد و ميدل وير خاصه بتصريح الدخول ان مين يدخل فى صفحه ما و مين ميدخلش**

يبقى بالمنطق كده المفروض ايه يحصل قبل ايه ؟؟

ادخل بالباسورد و الايميل و بعد كده لو انا يوزر عادى يبقى اروح للصفحه بتاعت اليوزر العادى و مرحش للصفحات بتاعت الادمن مثلا  
  
**عشان كده فيه حاجه فى الميدل وير اسمها Next و دى معناها لما تخلص الميدل وير دى و تمام التمام خد ال HttpRequest و دخله على الميدل وير الى بعدى و هكذا لحد ما يخلصوا :D**   
لو ميدل وير عملت ايرور فى النص بيخشش على الى بعديها و بيرجع تانى و مبيكملش ههههههههه

**خد بالك برضوا كتر الميدل وير مش حاجه حلوه للمشروع لان كل ريكويست لازم هيعدى على الميدل ويرز كلها و لو الميدل وير كتير هيبقى حاجه وحشه جدا و هتبطء الكود**

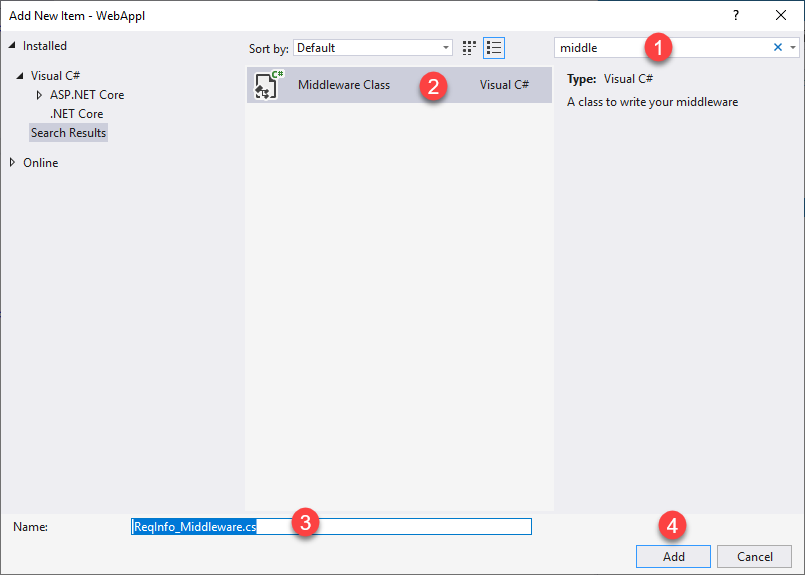
MB

################################################################

Custom Middlewares

------------------------------

أول حاجه روح اعمل كلاس Middleware



namespace WebAppl.Middlewares

{

// You may need to install the **Microsoft.AspNetCore.Http.Abstractions** package into your project

public class ReqInfo\_Middleware

{

private readonly RequestDelegate \_next;

**// Dependency Injection for Request Delegate to use it**

public ReqInfo\_Middleware(RequestDelegate next)

{

\_next = next;

}

public Task Invoke(HttpContext httpContext)

{

**// هنا بكتب الكود الخاص بالميدل وير و بعد كده الميدل وير يتعمله نيكست**

return \_next(httpContext);

}

}

// Extension method used to add the middleware to the HTTP request pipeline.

public static class ReqInfo\_MiddlewareExtensions

{

public static IApplicationBuilder UseReqInfo\_Middleware(this IApplicationBuilder builder)

{

return builder.UseMiddleware<ReqInfo\_Middleware>();

}

}

}

طيب دى قبل ما اعمل اى حاجه فى الميدل وير طب لو انا عاوز ان الميدل وير دى مجرد بس اعمل فيها لوجر بيعرفنى معلومات عن الريكويست و خلاص هكتب الكلام ده اذاى ؟

اول حاجه هعمل انجيكت للوجر اوبجيكت فى الكونستراكتور عشان استخدمه فى الكلاس هنا و بعد كده هروح على فانكشن ال Invoke اكتب الكود بتاعى الى انا عاوز اكتبه بقه و ادى الكود بعد التشغيل

namespace WebAppl.Middlewares

{

// You may need to install the Microsoft.AspNetCore.Http.Abstractions package into your project

public class ReqInfo\_Middleware

{

private readonly RequestDelegate \_next;

private readonly ILogger \_Logger;

public ReqInfo\_Middleware(RequestDelegate next , ILoggerFactory loggerFactory )

{

\_Logger = loggerFactory.CreateLogger<ReqInfo\_Middleware>();

\_next = next;

}

public async Task Invoke(HttpContext httpContext)

{

\_Logger.LogInformation("Req Path Info : " + httpContext.Request.Path);

// هنا بكتب الكود الخاص بالميدل وير و بعد كده الميدل وير يتعمله نيكست

await \_next.Invoke(httpContext);

\_Logger.LogInformation("MW Finshed");

}

}

// Extension method used to add the middleware to the HTTP request pipeline.

public static class ReqInfo\_MiddlewareExtensions

{

public static IApplicationBuilder UseReqInfo\_Middleware(this IApplicationBuilder builder)

{

return builder.UseMiddleware<ReqInfo\_Middleware>();

}

}

}

الكلاس الى تحت ده معمول عشان اعمل فيه Extention function من نوع ال IApplicationBuilder و ده عشان اعرف استخدم البيلدر بتاع الميدل وير ده فى ال Startup.cs و اناديه بالفانكشن UserReqInfoMiddleware

طيب بعد كده عشان اشغل الميدل وير دى بروح اناديها فين يا معلم ؟؟

بناديها فى ال Configration الى فى الStartup.cs

public void Configure(IApplicationBuilder app, IWebHostEnvironment env)

{

if (env.IsDevelopment())

{

app.UseDeveloperExceptionPage();

}

else

{

app.UseExceptionHandler("/Error");

// The default HSTS value is 30 days. You may want to change this for production scenarios, see https://aka.ms/aspnetcore-hsts.

app.UseHsts();

}

// Middlewares

app.UseHttpsRedirection();

app.UseStaticFiles();

app.UseRouting();

app.UseAuthorization();

**// Register Middleware Here**

**app.UseReqInfo\_Middleware();**

app.UseEndpoints(endpoints =>

{

endpoints.MapRazorPages();

});

}

**كده خلاص يا صديقى الميدل وير هتتنفذ على كل الريكويستات و لو تمام هتعمل للريكويست Next مش تمام هترد عليها و تقولها انتى مش تمام يا ريكويست :D**

**مثال للميدل وير هتفهم منه الدنيا عمليا**

دلوقتى فى ميدل وير خاص بتسجيل الدخول بيشوف اذا كان اليوزر نيم و الباسورد صح ولا لأ صح كده ؟؟

ايوه اهه الميدل وير ده فى كل مره انت بتيجى تعمل فيها ريكويست بيروح الريكويست معدى عليه و يعمل فحص هل اليوزر نيم و الباسورد دول بتوع الكلاينت ده فالسيشن متخزن فيها الكلام ده فتديهوله يروح شايف اليوزر نيم و الباسورد يلاقيهم صح يعديه و هكذا بقه فى كل ريكويست

طيب لو احنا مكناش مستخدمين السيشن و الكوكيز او اى حاجه ليها علاقه بال state management كان ايه الى هيحصل ؟؟

كان الى هيحصل ان كل مره هبعت ريكويست هيروح رايح معدى على الميدل وير و هيطلب مننا الباسورد و لازم اديهاله ههههههههه تخيل بقه يعنى من الاخر السيشن انقذتنا ههههههههههههه

**انا غيرت اللوجر فى الميدل وير عشان استخدم ال Nlog عشان استخدامه اسهل و عشان بيرمى على فايل بينما اللوجر العادى على الترمينال بس**

private readonly RequestDelegate \_next;

private readonly ILogger<ReqInfo\_Middleware> \_logger;

public ReqInfo\_Middleware(RequestDelegate next , ILogger<ReqInfo\_Middleware> logger)

{

\_logger = logger;

\_next = next;

}

public async Task Invoke(HttpContext httpContext)

{

\_logger.LogInformation("Req Path Info : " + httpContext.Request.Path);

// هنا بكتب الكود الخاص بالميدل وير و بعد كده الميدل وير يتعمله نيكست

await \_next.Invoke(httpContext);

\_logger.LogInformation("MW Finshed");

}

MB

################################################################

Model and Data Folder

----------------------------------

ال Model ده ده عباره فولدر بحط فيه كل الكلاسات الى انا عاوز استخدمها للتعامل مع نتايج مختلفه عن كلاس ال Data الى بيتعمل مع الداتا بيز بشكل مباشر   
بمعنى   
دلوقتى انا عندى داتا بيز المفروض ان فى جدول و الجدول ده فيه Columns كل كولوم من دول بيبقى فى كلاس ال Data او ال Entity بيبقى عامل Mapping مع Propriety فى كلاس ال Data و اى حاجه بتختلف فى المابينج الدنيا بتبوظ فلازم تبقى واخد بالك و انت بتعمل مابينج و فى كلاسات ال Data , Entity مينفعش تزود اى بروبرتى عن الموجود فى الجدول   
  
فمثلا و ليكن انا دلوقتى عاوز ارجع حاجه مش موجود فى الانتيتى او الداتاالمفروض ساعاتها الحاجه الى هرجعها تبقى بترجع بريتيرن من كلاس جديد خالص يبقى فيه حاجات الانتيتى و زياده الحاجات الجديده الى انا محتاجها و من هنا ظهر فكره ال ViewModel  
  
تعالى بقه نعمل كلاسات الداتا و المودل و اوريك استخدامهم   
اولا انا هعمل ال Data بس انا مش هعملها فى فولدر انا هعملها فى مشروع تانى خالص اسمه web.Data و يكون بيكلم داتا بيز بال Entity كمان   
هروح على ال Solution و ادوس كليم يمين add ثم New Project و هضيف ال Entity بقه

################################################################

EFW 6 VS EFW Core

---------------------------------------------

الفرق بين الانتيتى فريم ورك الكور و 6 ان الكور تدعم ال Crossplatform و شويه حاجات تانيه كده بس كده شكرا باقى الحاجات هما الاتنين واحد

و بالتالى لما احب اشتغل كروس بلات فورم هتجه للدوت نت كور وش انما لو مش عاوز كروس بلات فورم يبقى هتعامل مع العاديه و خلاص  
اهميه ال Entity Framework انها بتربط بين ال model و الداتا بيز بسهوله جدا لانها بتعمل مابينج ما بين الكود و الداتا بيز كويس جدا

MB

################################################################

Install EFW Core

----------------------------------

**اذاى تسطب ال Entiy على مشروعك و خد بالك ان EFW 5 , 6 دول حاجه تانيه خالص غير ال EFW Core و بالتالى فى اختلاف كبير فى كلاساتهم و طالما هتسطب الكور عشان الكروس بلاتفورم و الكلام ده يبقى خلاص انسى بقه شغل ال EFW 5 , 6 خالص   
احنا هنركز على اننا نشتغل بال EFW Core فقط**

**بص يا سيدى امشى ذى الخطوات دى بالظبط ده لشرح الانتيتى كله حاجه جميله و فيه اذاى تسطبه من الاول خالص**

<https://www.entityframeworktutorial.net/efcore/entity-framework-core.aspx>

ثم ده

<https://www.devart.com/dotconnect/postgresql/docs/EFCore-Database-First-NET-Core.html>

ثم ده عشان تعمل الداتا بيز فيرست لما اجى اعوز اعمل كريشن للكود من الدتا بيز اعرف اهمله بحاجه اسمها

## Scaffold-DbContext

من هنا

<https://docs.microsoft.com/en-us/ef/core/miscellaneous/cli/powershell#scaffold-dbcontext>

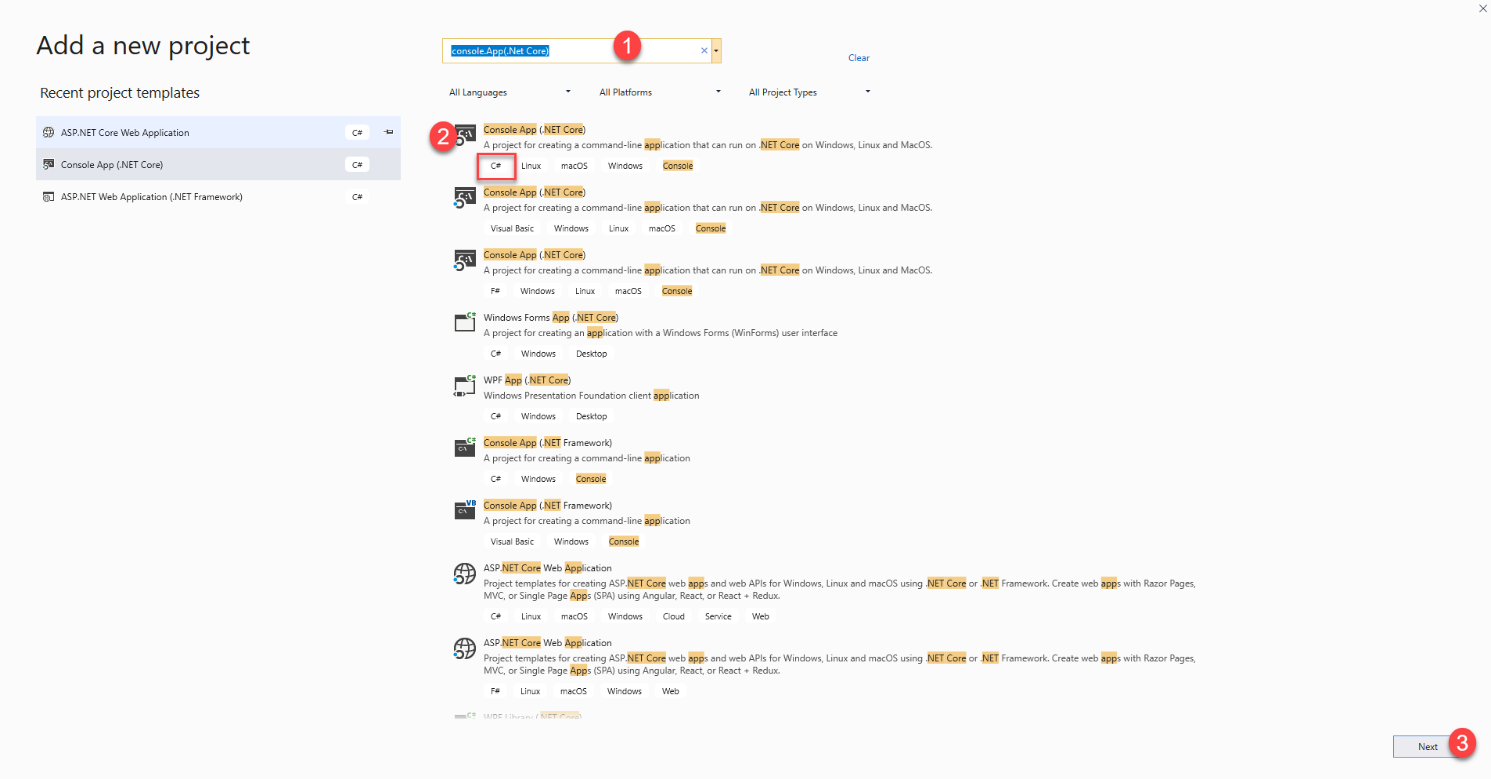
**Scaffold-DbContext "Server=.;Database=SPSDB;Trusted\_Connection=True;" Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models**

**ده الامر ده ذى ما انت شايف بيعممل الكلاسات مجرد بتديله السيرفر و اسم الداتا بيز و الفولدر الى يحطلك الكلاسات فيها و كده و عمتا اقرأ فيه من هنا**

<https://www.entityframeworktutorial.net/efcore/create-model-for-existing-database-in-ef-core.aspx>

طيب خلاص انا هشتغل Code First كده الدنيا هتبقى سهله و جميله خالص ان شاء الله   
**بس عمتا هتشتغل Code First أو DB First   
أول حاجه هتعمل مشروع جديد فى السوليوشن Solution من نوع**

**Console App (.Net Core) من نوع ال C#**

****

**و بعد كده هسطب فى المشروع بتاع ال و entity و كذلك فى مشروع ال API الباكيدجات الاتيه**

**Install-Package Microsoft.EntityFrameworkCore.Tools**

**Install-Package Microsoft.EntityFrameworkCore**

**Install-Package Microsoft.EntityFrameworkCore.SqlServer**

كده خلاص يا باشا اقدر استخدمه DB First بال scaffold Db-context او Code First عن طريق انى اعمل كلاس الكونتيكست و كلاسات الجداول و اعملها ميجريشن بقه يعنى انه يعمل كل حاجه و يعملها ميجريشن عادى بقه فبما انى شغلت

ال DB First دلوقتى فتعالى نجربه بال scaffold Db-context هتلاقى انه عمل ال SPSDBContext بالشكل ده

using System;

using Microsoft.EntityFrameworkCore;

using Microsoft.EntityFrameworkCore.Metadata;

namespace Web.Entity.Models

{

public partial class SPSDBContext : DbContext

{

public SPSDBContext()

{}

public SPSDBContext(DbContextOptions<SPSDBContext> options)

: base(options)

{}

public virtual DbSet<Cities> Cities { get; set; }

public virtual DbSet<Persons> Persons { get; set; }

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

{

if (!optionsBuilder.IsConfigured)

{ optionsBuilder.UseSqlServer("Server=.;Database=SPSDB;Trusted\_Connection=True;");

}

}

protected override void OnModelCreating(ModelBuilder modelBuilder)

{

modelBuilder.Entity<Cities>(entity =>

{

entity.Property(e => e.Id)

.HasColumnName("ID")

.ValueGeneratedNever();

entity.Property(e => e.Name).HasMaxLength(50);

});

modelBuilder.Entity<Persons>(entity =>

{

entity.Property(e => e.Id)

.HasColumnName("ID")

.ValueGeneratedNever();

entity.Property(e => e.Name).HasMaxLength(50);

});

OnModelCreatingPartial(modelBuilder);

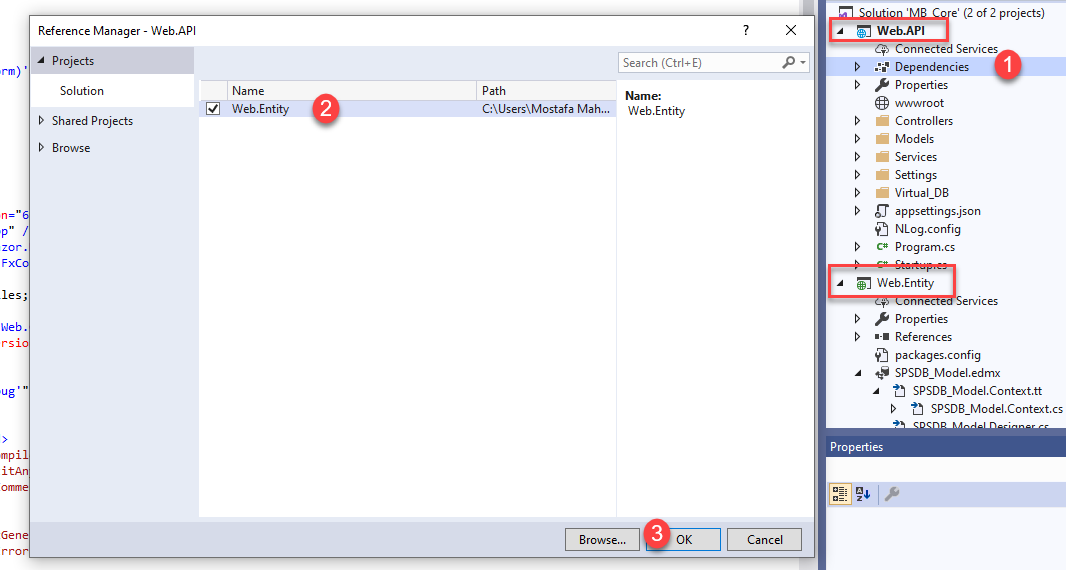
}

partial void OnModelCreatingPartial(ModelBuilder modelBuilder);

}

}

طيب انا عاوز اربط المشروع Entity بمشروع ال API عشان اقدر استخد ال Entity جوه مشروع ال API و اتعامل مع الداتا بيز من الكنترولار بشكل مباشر يبقى المفروض اخد ريفرنس من مشروع ال Entity احطه فى مشروع ال API

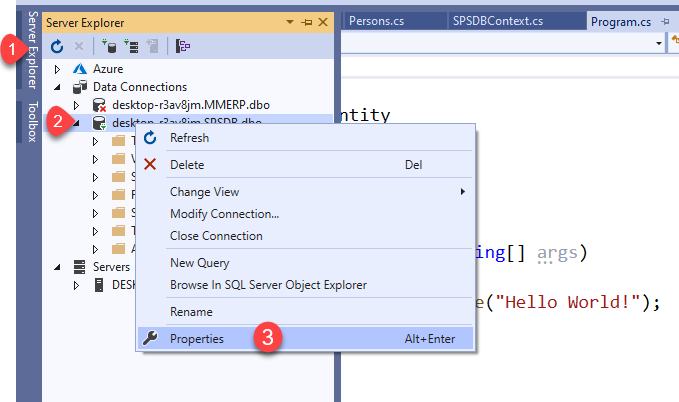


طيب دلوقتى انا عاوز اجرب الانتيتى اشوفه اشتغل فعلا فى مشروع ال API ولا لأ فأول حاجه هتأكد انى مسطب على مشروع ال API الباكيدجات الاتيه

**Install-Package Microsoft.EntityFrameworkCore.Tools**

**Install-Package Microsoft.EntityFrameworkCore**

**Install-Package Microsoft.EntityFrameworkCore.SqlServer**

بعد كده هروح على ال StartUp و اتأكد انى رابط مشروع ال API بنفس الكونيكشن بتاع الداتا بيز الى انا رابطه مع الانتيتى   
انا كده كده بروح اجيب الكونيكشن من الحته دى   


هاخد الكونيكشن سترينج ده و اروح احطه فى ال AppSettings.json

{

"ConnectionString": {

"DefaultConnStr": "Data Source=.;Initial Catalog=SPSDB;Integrated Security=True"

},

"Logging": {

"LogLevel": {

"Default": "Information",

"Microsoft": "Warning",

"Microsoft.Hosting.Lifetime": "Information"

}

},

"AllowedHosts": "\*"

}

بعد كده هقراه فى ال StartUp.cs فى فانكشن الكونفيجريشن سيرفيسيس

public void ConfigureServices(IServiceCollection services)

{

var ConnString = Configuration.GetSection("ConnectionString")["DefaultConnStr"];

services.AddDbContext<SPSDBContext>(options => options.UseSqlServer(ConnString));

services.AddControllers();

}

كده خلاص بقى مرتبط بالداتا بيز هو كمان   
طيب دلوقتى انت ربطت المشروع API بالكونيكشن بتاع الداتا بيز و كنت قبل كده واخد ريفيرنس جواه من مشروع ال Entity اذا انت جوه الكنترولار تقدر تنادى ال Entity و تتعامل مع الداتا بيز عن طريق ال context

namespace Web.API.Controllers

{

[ApiController]

[Route("[controller]")]

public class WeatherForecastController : ControllerBase

{

private readonly ILogger<WeatherForecastController> \_logger;

private readonly SPSDBContext \_Mycontext;

public WeatherForecastController(ILogger<WeatherForecastController> logger , SPSDBContext Mycontext )

{

\_logger = logger;

\_Mycontext = Mycontext;

}

[HttpGet]

public IActionResult Get()

{

var Data = \_Mycontext.Cities.ToList();

return Ok(Data);

}

}

}

وهبعت ريكويست بقه كده و اجرب المفروض انه كده هيرجع داتا بقه من الداتا بيز   
<https://localhost:44304/weatherforecast>

كده ال Entity شغال جوه ال API يبقى ناقص انى اظبط حاجه بسيطه و هى انا بدل ما كل مره بقه هعد اعمل ديبندينسى انجيكشن للانتيتى جوه الكونستراكتور فانا منا ممكن اعمل الكلام ده بطريقه حلوه و هى انى اعمل ApiBaseController و يبقى الكلاس ده بيورث من الكلاس ControllerBase و اضيف ليه كل الى هحتاجه فى كل الكنترولارز بدل ما اكتبها كل مره هكريت كنترولار جديد

يبقى يابرنس اول حاجه هتعملها كلاس اسمه كنترولار بيورث من الكلاس الاساسى لل API الى هو ControllBase عشان يحمل كل خواصه   
و طبعا هضيف ليه ال Logger و ال Context عشان موجعش دماغى كل شويه احط الحاجه و اقرف نفسى   
اهه ده بقه ال APIBaseController.cs وذى ما انت شايف بيورث من اهم كلاس فى الويب ايه بى اى الكنترولار بيس و عدل عليه و زود الكونتيكست و اللوجر

public class ApiBaseController : ControllerBase

{

protected readonly ILogger<WeatherForecastController> \_logger;

protected readonly SPSDBContext \_Mycontext;

public ApiBaseController(ILogger<WeatherForecastController> logger, SPSDBContext Mycontext)

{

\_logger = logger;

\_Mycontext = Mycontext;

}

}

طيب دلوقتى بقه كل اما هاجى استخدم اى كنترولار جديد انا مش مضطر اعمل ال Dependency Injection من اول و جديد لا ده مجرد بعمل للكلاس انه يورث من ال APIBaseController و ابعتله اتنين برامتيار فى الكونستراكتور فى ال Base عشان يروح يجيب بتوع ال Base Class فى الكنترولار الى انا فيه فى الوقت الحالى فيبقى شكل الكنترولار الحالى كده

namespace Web.API.Controllers

{

[ApiController]

[Route("[controller]")]

public class WeatherForecastController : ApiBaseController

{

public WeatherForecastController(ILogger<WeatherForecastController> logger, SPSDBContext Mycontext) :base(logger , Mycontext) { }

[HttpGet]

public IActionResult Get()

{

var Data = \_Mycontext.Cities.ToList();

return Ok(Data);

}

}

}

و كده خلاص تقدر تستخدم الكونتيكست ياعم من غير ما تعملهم DI هنا لا هو مجرد ذى ما انت شايف بس

**اسم الانتيتى خليه فيه كلمه Context دايما فيبقى كده SPSDB\_Context**

**نصيحه صحيح لو لقيت ان ال ِ ADO.Net مش بيظهرلك فى المشاريع بعد ما انت عملت الخطوات الى فوق دى يبقى تروح تتاكد انك مسطب ال Entity Tools**

و ده تلخيص جميل للمشاكل الى بتحصل و انت لسه بتكريت مشاريع اهه و حلها   
<https://stackoverflow.com/questions/44299332/cant-find-ado-net-entity-data-model-template-in-vs2017>

**خد بالك ان ال App.Config ده مش هيظهرلك فى المشروع بتاع ال Core الا لما تسطب ال**

EntityFramWorkCore

EntityFramWorkCore.SqlServer

فى مشروع ال Wep.APl او ال Entity لو محتاجه و متعاملش اعمله مانوال ياعم او سيرش اذاى تعمله

ممكن تبص على دى بيعملك فيها اذاى تعمل ال App.config   
<https://stackoverflow.com/questions/45034007/using-app-config-in-net-core>  
<https://docs.microsoft.com/en-us/visualstudio/ide/how-to-add-app-config-file?view=vs-2019>

MB

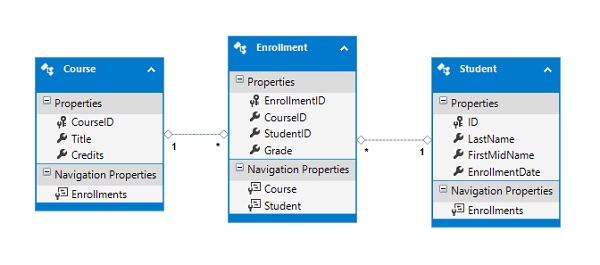
################################################################

EF Code First

----------------------------------

نفس كل الخطوات الى فاتت فى ال DB First بس الفرق ان انا هنا هعمل الكونتيكست و كلاسات الجداول بتاعت الداتا بيز بأيدى و بعد ما هخلصها هعمل ميجريشن للحاجات دى بحيث تترفع على الداتا بيز و استخدمها بقه   
بص على اهم حاجه اللينك ده جايب من الاخر   
<https://www.entityframeworktutorial.net/efcore/entity-framework-core-console-application.aspx>

و ليكن بقه انت عاوز تعمل المودل ده تحوله لكلاسيس و من الكلاسيس يتعمل داتا بيز بقه



اول حاجه طبعا لازم اروح اعمل لكل جدول فيهم كلاس

على فكره الجدول ده عباره عن جدولين عاملين مع بعض علاقه Many to Many فاضروا يعملوا جدول ال Enrollment عشان يبقى جدول وسيط بينهم

تعالى بقه نحول كل واحد يا كبير

اول حاجه هنبدا بال Student فهعمل كلاس منفصل ليه باسمه

namespace CodeFirst\_App.TablesClasses

{

public class Student

{

// Properties for Table

[Key]

public int ID { get; set; }

public string LastName { get; set; }

public string FirstMidName { get; set; }

public DateTime EnrollmentDate { get; set; }

// Table Navigations for Another Table

public virtual ICollection<Enrollment> Enrollments { get; set; }

}

}

ذى ما انت شايف بحط ال columns الى فى الجدول و بعد كده بحط كوليكشن من الكلاس الى المفروض ال student مربوط معاه ربطه one-to-Many و انت بتعرف الكلاس الى هتربط بيها الجدول من حاجتين اولهم الروابط فى الى فى الصوره و ثانيا مكتوب اصلا ان الجدول بيعمل Navigate على جدول تانى اسمه Enrollment

و طبعا Key دى يعنى الاى دى Primary Key و ده اتربييوت من ال DataNotation و ال Schema

كذلك برضوا هعمل الCourse بنفس الفكره

namespace CodeFirst\_App.TablesClasses

{

public class Course

{

// Properties for Table

[Key]

public int CourseID { get; set; }

public string Title { get; set; }

public int Credits { get; set; }

// Table Navigations for Another Table

public virtual ICollection<Enrollment> Enrollments { get; set; }

}

}

بعد كده بقه اعمل الكلاس الى اسمه Enrollment

namespace CodeFirst\_App.TablesClasses

{

public enum Grade

{

A=1, B=2, C=3, D=4, F=5

}

public class Enrollment

{

[Key]

public int EnrollmentID { get; set; }

public int CourseID { get; set; }

public int StudentID { get; set; }

public Grade? Grade { get; set; }

[ForeignKey("CourseID")]

public virtual Course Course { get; set; }

[ForeignKey("StudentID")]

public virtual Student Student { get; set; }

}

}

ذى ما انت شايف بما ان لكل طالب عدد كبير من ال Enrollments فحطينا ال Enrollments فى الكلاس بتاع ال student عباره عن Collection و كذلك فى الكلاس بتاع ال Course عملنا نفس الكلام

بس لما جينا بقه بنعمل كلاس ال Enrollment فالطبيعى بقه ان لكل عدد كبير من ال Enrollments هو طالب واحد فروحنا عاملينا هنا الرابطه بأوبجيكت واحد من الكلاس و مستخدمناش طبعا الكوليكشنز هنا

برضوا لازم تاخد بالك ان علامه الاستفهام (؟) معناها ان ال Column ده يقبل ال null و حاجه تانيه انا هنا عملت ال Grades فى Enum عشان مش عااوز اربط الدرجات بال ABCEF بس

و طبعا CourseID دى ForiegnKey هى و ال ID واحد للكورسات و واحد للطلبه

كده تكوين ال Classes تم و خد بالك ان الFK لازم يبقى اسم البروبرتى هو هو نفس الاسم الى انت عامله FK ك Data Anotation

ناقص بقه انى اعمل ايه ؟؟

انى اربط المشروع بال Database و اعرف المشروع و الداتا بيز بالجداول الى هيتربط بيها عن طريق ال DataSet

فهعمل كلاس و هسميها ال MyContext و هيكون بيورث من الكلاس الى اسمه DbContext عشان ياخد خواصه و يزود عليه الى انا عاوزه بقه ههههههه

namespace CodeFirst\_App

{

public class SPSDBContext: DbContext

{

public SPSDBContext(){}

public SPSDBContext(DbContextOptions<SPSDBContext> options): base(options) {

// Database.EnsureCreated();

}

public virtual DbSet<Course> Courses { get; set; }

public virtual DbSet<Enrollment> Enrollments { get; set; }

public virtual DbSet<Student> Students { get; set; }

}

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

{

if (!optionsBuilder.IsConfigured)

{

optionsBuilder.UseSqlServer("Server=.;Database=SPSDB;Trusted\_Connection=True;");

}

}

protected override void OnModelCreating(ModelBuilder modelBuilder)

{ }

partial void OnModelCreatingPartial(ModelBuilder modelBuilder);

}

كده بقى مش باقى غير انى اعمل الكونيكشن سترينج

طب و دى اعملها اذاى ؟؟ بيقولك يا سيدى انت تخش تعمل كونيكشن بين البرنامج و الداتا بيز على اى داتا بيز موجوده و كل الى هتعمله انك هتغير اسم الداتا بيز بالحاجه الى انت عاوزها بقه

فى طرق كتير منها انك تعمل فايل تسميه badawy.udl



هتختار فى البروفايدر : Microsoft OLE DB Provider for SQL Server

بعد كده فى الكونيكشن هتمشى ذى ما هو مكتوب تحط اسم السيرفر و تدخل الكردينشيالز و تختار اى داتا بيز انت عارفها بعد كده هتفتح الملف بتاع ال udl ده كأنه Text هتلاقى مكتوب جواه ال ConnectionString هتاخده من اول Integrated Security لحد الاخر و شكرا على كده

و ممكن اصلا تعمل كونيكشن عن طريق ال tools فى ال Visual Studio و تاخد منها ال ConnectionString برضوا و فى الاخر بقه خد الكونيكشن سترينج ده و حطه فى ال

optionsBuilder.UseSqlServer("Server=.;Database=SPSDB;Trusted\_Connection=True;");

دلوقتى بقه المفروض تعمل ايه ؟؟

اعمل Migration

طب و ده يتعمل اذاى ؟؟

أول حاجه هتفتح ال Package Manager Console

بعد كده هتنفذ الامر ده عشان تشغل حوار ال Migration ده   
**Enable-Migrations**  
و لازم تتأكد انك على البروجيكت الصح

يا اما تعمل الكوماند كامل كده

Enable-Migrations -ProjectName CodeFirst\_App -ContextTypeName MyContext

كده هيتعمل بقه فايل Configuration خاص بال CodeFirst يا كبير

كلمه ميجريشن دى يعنى الكلاسيز الى انت عاملها للانتيتى تتحول لداتا بيز

طيب نيجى بقه بعد ما شغلنا الميجريشان عاوزين ننشأ الداتا بيز بقه بالكلاسيس الى انا عملتها هعمل ايه بقه ؟؟

هنفذ الامر ده بقه الى هو بقوله ضيفلى الميجريشن الى اسمها كود فيرشن واحد

**Add-Migration CodeFirst-Version-1**

هتلاقى ان فيه فايل اتعمل و اسمه بالاسم الى انت عامله للميجريشن و فيه بقه الحاجات الى هتحصل Down , Up

ال Down دى بيعمل فيها دروب للحاجات الى انت عاملها فى حاله انك رجعت فى كلامك و مش عاوز الفرشن دى

و ال Up ده بينفذ فيها الحاجه الى انت هتعملها جديده فى الاستراكتشار

طيب بعد ما شوفت الى مكتوب فى الفايل و عاوز تخلى الى انت عملته يبقى موجود فى الداتا بيز يبقى تقوله ياباشا رن الكوماند ده

**update-database**

طب لو عاوز فيرشن معينه من الميجريشن يا اسطى يبقى تترجيت الفرش الى انت عاوزها دى باسمها

**update-database -TargetMigration:CodeFirst-Version-1**

روح بص بقه على الداتا بيز شوف ايه الى جرى هتلاقى الداتا بيز تم انشائها

و بص هنا هتفهم كتير

لو احتجت تعرف اى معلومات عن ال Commands هتكتب قبل الكوماند جمله get help

ذى كده

get-help add-migration

لو عاوز بقه الميجريشن يبقى اوتوماتيك بس منصحش بيه بصراحه يبقى رن الكوماند ده

enable-migrations –EnableAutomaticMigration:$true

طيب عملت بقه تغيير بسيط و عاوز ازود column اسمه Code فى ال Courses هروح اضيفه و اعمل ميجريشن جديد باسم جديد بقه

namespace CodeFirst\_App.TablesClasses

{

public class Course

{

// Properties for Table

[Key]

public int CourseID { get; set; }

public string Title { get; set; }

public int Credits { get; set; }

**public string Code { get; set; }**

// Table Navigations for Another Table

public virtual ICollection<Enrollment> Enrollments { get; set; }

}

}

ثم

Add-Migration CodeFirst-Version-2

ثم

update-database

هيحصل تحديث فى الجدول و مفيش داتا هتتمسح منه لا تقلق

طيب فى حاجه كده لازم اقولهالك بقه

لو انت عاوز تعمل ال PK , FK و تغيير اسامى الاعمده فى الجداول ممكن بدل ما تعملها API تعملها عن طريق الكونفيجريشن الى هى Fluent API و دى بتتكت جوه فانكن داخل ال DbContext و ده مثال لتغيير اسم عمود فى جدول ال student

public class MyContext : DbContext {

**protected override void OnModelCreating(**ModelBuilder **modelBuilder) {**

**modelBuilder.Entity<Student>().Property(s ⇒ s.FirstMidName)**

**.HasColumnName("FirstName");**

**}**

public virtual DbSet<Course> Courses { get; set; }

public virtual DbSet<Enrollment> Enrollments { get; set; }

public virtual DbSet<Student> Students { get; set; }

}

من FirstMidName الى FirstName

الكود فيرست عمتا حاجه ممتعه جدا جدا و للعلم الداتا بيز مش بتقع لما تبعمل اى تغيير مهما كان الا لو انت عاوز تدروب حاجه فعلا فبتمسحها من الداتا سيت او تمسح عمود من جدول ساعتها العمود بيتمسح بس بيسيب باقى الداتا عادى و دى حاجه غبقريه جدا الكود فيرست ده من الاخر اقسم بالله اختراع و عشان تتعمق فيه بزياده بقه تعايش هنا

<https://www.entityframeworktutorial.net/code-first/simple-code-first-example.aspx>

اما لو عاوز تفهم فى السريع كده يبقى من هنا

<https://www.tutorialspoint.com/entity_framework/entity_framework_code_first_approach.htm>

دى بقه لو انت عاوز تعمل كل الى انت عامله Data Annotation عن طريق ال Fluent API

اول حاجه بتعمل كل كلاس لحدها كده و دى كلاسات خفيفه كده عشان مكتبش كود كبير يعنى

Person.cs

public partial class Persons

{

public int Id { get; set; }

public string Name { get; set; }

public string Address { get; set; }

}

Cities.cs

public partial class Cities

{

public int Id { get; set; }

public string Name { get; set; }

public string Description { get; set; }

}

SPSDBContext.cs

public partial class SPSDBContext : DbContext

{

public SPSDBContext()

{

}

public SPSDBContext(DbContextOptions<SPSDBContext> options)

: base(options)

{

//Database.EnsureCreated();

}

public virtual DbSet<Cities> Cities { get; set; }

public virtual DbSet<Persons> Persons { get; set; }

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

{

if (!optionsBuilder.IsConfigured)

{

optionsBuilder.UseSqlServer("Server=.;Database=SPSDB;Trusted\_Connection=True;");

}

}

//Fluent API By Model Builder

protected override void OnModelCreating(ModelBuilder modelBuilder)

{

modelBuilder.Entity<Cities>(entity =>

{

entity.Property(e => e.Id)

.HasColumnName("ID")

.ValueGeneratedNever();

entity.Property(e => e.Name).HasMaxLength(50);

});

modelBuilder.Entity<Persons>(entity =>

{

entity.Property(e => e.Id)

.HasColumnName("ID")

.ValueGeneratedNever();

entity.Property(e => e.Name).HasMaxLength(50);

});

OnModelCreatingPartial(modelBuilder);

}

partial void OnModelCreatingPartial(ModelBuilder modelBuilder);

}

هنل عملت داخل ال context ال fluent Api و قولتله ايه خصائص كل كولوم و لو عاوز ارعفه ايه ال key و ايه ال FK اديله بيانات اكتر عن كل كولوم هعمل عادى فى ال ModelBuilder لان ده بياخد الكود يحوله لحاجات خاصه بالجداول فى الداتا بيز   
و فى الكونتيكست برضوا ذى ما انت شايف عرفته الكونيكشن استرينج و ال DatSet الى هى الجداول الى هنتعمل معاها و كل حاجه   
عمتا انا بحب طريقه ال Data Annotation فى معظم الحاجات و الكونفيرجيشن دى قت اللزوم و بس

بعد كده بعمل الاوامر الاتيه عشان يعمل ميجريت و ينشأ الداتا بيز او يعدل فيها   
أول حاجه هتفتح ال Package Manager Console

بعد كده هتنفذ الامر ده عشان تشغل حوار ال Migration ده   
**Enable-Migrations**  
و لازم تتأكد انك على البروجيكت الصح

يا اما تعمل الكوماند كامل كده

Enable-Migrations -ProjectName Web.Entity\_App -ContextTypeName SPSDBContext

هنفذ الامر ده بقه الى هو بقوله ضيفلى الميجريشن الى اسمها كود فيرشن واحد

**Add-Migration CodeFirst-Version-1**

طيب بعد ما شوفت الى مكتوب فى الفايل و عاوز تخلى الى انت عملته يبقى موجود فى الداتا بيز يبقى تقوله ياباشا رن الكوماند ده

**update-database**

**ممكن بدل ما تعمل اخر امر بتاع ال update-database تعمله انه يتعمل ابديت للداتا بيز مع أول Run طب و دى تتعمل اذاى اصديقى ؟؟  
تتعمل فى فايل ال program بكل بساطه هقوله و انت بتعمل Build للبروجيكت اعملى ميجريت للداتابيز**

public class Program

{

public static void Main(string[] args)

{

**var Host = CreateHostBuilder(args).Build();**

**using (var scope = Host.Services.CreateScope())**

**{**

**try**

**{**

**var context = scope.ServiceProvider.GetService<SPSDBContext>();**

**context.Database.Migrate();**

**}**

**catch (Exception ex)**

**{**

**throw ex;**

**}**

**}**

**Host.Run();**

}

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

Host.CreateDefaultBuilder(args)

.ConfigureWebHostDefaults(webBuilder =>

{

webBuilder.UseStartup<Startup>();

});

}

كده كل اما هتعمل Add لاى ميجريشن جديد هيتعمله update فى ال Run time

لو هعمل الكلام ده بال Data Annotation هيبقى كده

Cities.cs

public partial class Cities

{

[Key]

public int Id { get; set; }

public string Name { get; set; }

public string Description { get; set; }

}

Persons.cs

public partial class Persons

{

[Key]

public int Id { get; set; }

public string Name { get; set; }

public string Address { get; set; }

}

SPSDBContext.cs

namespace Web.Entity.Models

{

public partial class SPSDBContext : DbContext

{

public SPSDBContext()

{}

public SPSDBContext(DbContextOptions<SPSDBContext> options)

: base(options){ }

public virtual DbSet<Cities> Cities { get; set; }

public virtual DbSet<Persons> Persons { get; set; }

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

{

if (!optionsBuilder.IsConfigured)

{

optionsBuilder.UseSqlServer("Server=.;Database=SPSDB;Trusted\_Connection=True;");

}

}

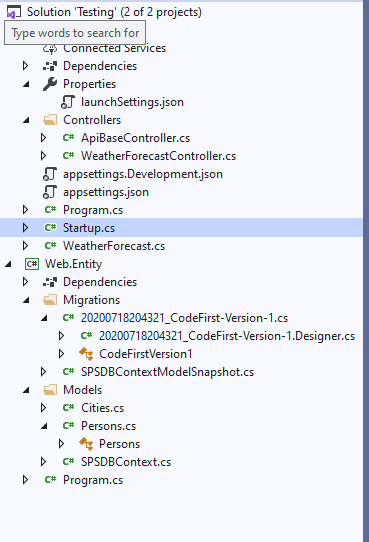
protected override void OnModelCreating(ModelBuilder modelBuilder)

{}

partial void OnModelCreatingPartial(ModelBuilder modelBuilder);

}

}

نعمل بقه add للميجريشن و update للداتا بيز كده كل الكلام ده تمام يا ريس   
صوره صحيح للفايلات و البروجيكتات عشان تبقى شايف الشكل الجميل   


MB

##########################################################################

Seeding Data to Database in Code first

----------------------------------------------------------

بتروح على ال context هتلاقى فانكشن اسمها OnModelCreating دى دواها بقه بتحط ال Dummy data الى انت عاوز تملى الجداول بيها بالطريقه الى هتشوفها دى و اول ما بتخلص بتعمل add للميجريشن و **update للداتا بيز بس عن طريق الكود بتاع الميجريشن فى الRun Time تلاقى البيانات الكنافه دى اترفعت على الداتا بيز فى الجداول انما لو جربت تعمل update للداتا بيز بطريقه عاديه مش هيرضى و هيبوظ**

public partial class SPSDBContext : DbContext

{

public SPSDBContext()

{}

public SPSDBContext(DbContextOptions<SPSDBContext> options)

: base(options){ }

public virtual DbSet<Cities> Cities { get; set; }

public virtual DbSet<Persons> Persons { get; set; }

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

{

if (!optionsBuilder.IsConfigured)

{

optionsBuilder.UseSqlServer("Server=.;Database=SPSDB;Trusted\_Connection=True;");

}

}

protected override void OnModelCreating(ModelBuilder modelBuilder)

{

modelBuilder.Entity<Persons>().HasData(

new Persons()

{

Id = 1,

Name = "Mahmoud Badawy",

Address = "Cairo"

},

new Persons()

{

Id = 2,

Name = "Mostafa Mahmoud",

Address = "Alex"

}

);

modelBuilder.Entity<Cities>().HasData(

new Cities()

{

Id = 1,

Name = "Mahmoud Badawy",

Description = "Lolololoy"

},

new Cities()

{

Id = 2,

Name = "Mostafa Mahmoud",

Description = "Lolololoy"

}

);

base.OnModelCreating(modelBuilder);

}

partial void OnModelCreatingPartial(ModelBuilder modelBuilder);

}

روح شوف الداتا بيز بقه هتلاقيه اتعملت و اتملت بالبيانات الى انت عاملها دى

MB

################################################################

Store Connection String Securely

----------------------------------------------------------

بيقولك ان تسجيل الكونيكشن سترينج فى ال appsettings.json غلط خصوصا لما يكون بيحتوى على رقم باسورد و اسم سيرفر ده حاجه غلط و المفروض يسجله بهيراركى معين فى الكونفيجريشن بس ببيانات غلط ذى كده

{

"ConnectionString": {

"DefaultConnStr": "HHHHHH Mafesh yad"

},

"Logging": {

"LogLevel": {

"Default": "Information",

"Microsoft": "Warning",

"Microsoft.Hosting.Lifetime": "Information"

}

},

"AllowedHosts": "\*"

}

و يروح رايح على البروبرتى بتاعت البروجيكت و يعمله الصح بقه بالشكل ده الى هو بيبقى بنفس الهيراركى بتاعه فى الكونفيجريشن و لان المكان ده بيعمل overwrite على كل الكونيفجريشن يا اسطى فكده انا اجبرته ياخد الكونيشكن من هنا و بكده يا كبير انا خليت الكونيكشن سترينج محدش يعرف عنه حاجه خلاص لانه بقى متشفر كده جوه البروجيكت و اول ما هتطلع بابليش هيبقى الكويرى سترينج متشفر و ده الصح لان لازم المعلومات المهمه دى تبقى متشفره و مش معروفه   
ملحوظه عشان حتى محدش يعرف الهيرارى الى انت عامل بيه امسح الكونيكشن سترينج خالص من ال AppSetting.json يبقى كده

{

"Logging": {

"LogLevel": {

"Default": "Information",

"Microsoft": "Warning",

"Microsoft.Hosting.Lifetime": "Information"

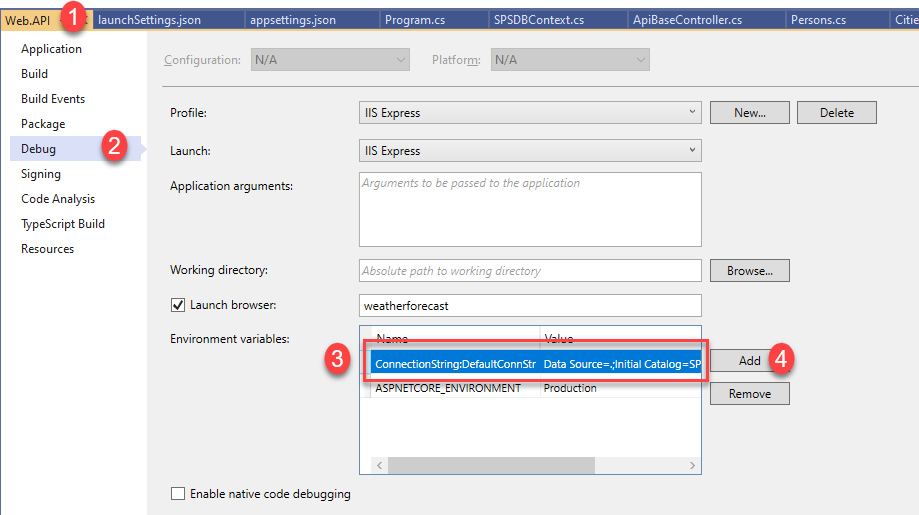
}

},

"AllowedHosts": "\*"

}

و اعمل نفس الهيراركى او بمزاجك فى البروبرتى ذى الصوره دى كده بالظبط و روح اقراه من الكونيفيجريشن بالكود



الصفحه الى انت شايفها جى هى هى ال launchSettings.json الى لو حبيت تغيير فى المنطقه الى انت بتغيير فيها دى هتروح تغيير فى البروفايل

{

“iisSettings”: {

“windowsAuthentication”: false,

“anonymousAuthentication”: true,

“iisExpress”: {

“applicationUrl”: “http://localhost:61559”,

“sslPort”: 44304

}

},

“$schema”: “http://json.schemastore.org/launchsettings.json”,

“profiles”: {

“IIS Express”: {

“commandName”: “IISExpress”,

“launchBrowser”: true,

“launchUrl”: “weatherforecast”,

“environmentVariables”: {

**“ConnectionString:DefaultConnStr”: “Data Source=.;Initial Catalog=SPSDB;Integrated Security=True”,**

**“ASPNETCORE\_ENVIRONMENT”: “Production”**

}

},

“Web.API”: {

“commandName”: “Project”,

“launchBrowser”: true,

“launchUrl”: “weatherforecast”,

“environmentVariables”: {

“ASPNETCORE\_ENVIRONMENT”: “Development”

},

“applicationUrl”: “https://localhost:5001;http://localhost:5000”

}

}

}

هتقرأ الكونيكشن سترينج الى موجود فى ال launchSetting.json اذاى ؟؟  
بطريقه من الاتنين دول و خد بالك انت ممكن تعمل الاسم بتاع الكونيكشن سترينج من غير هيراركى فى البروبرتى و تقراه من غير هيراركى بس الهيراركى بيديه حمايه اكتر

public void ConfigureServices(IServiceCollection services)

{

//var ConnString = Configuration.GetSection("ConnectionString")["DefaultConnStr"];

var ConnString = Configuration["ConnectionString**:**DefaultConnStr"];

services.AddDbContext<SPSDBContext>(options => options.UseSqlServer(ConnString));

services.AddControllers();

}

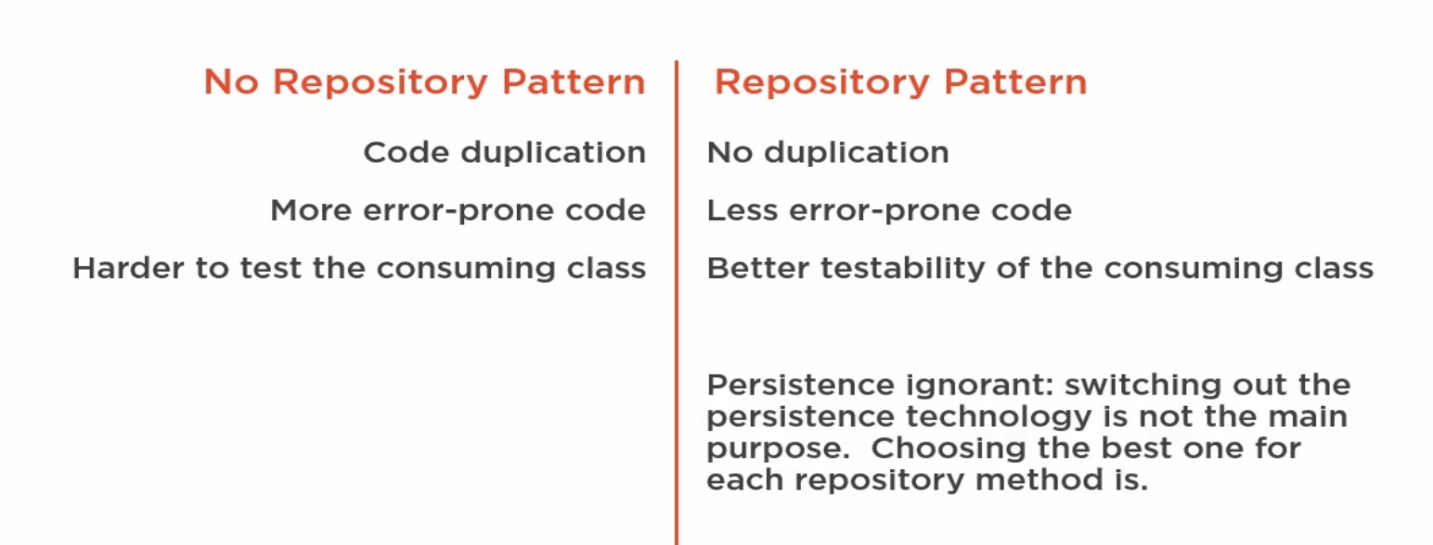
MB

################################################################

Repository Pattern

----------------------------------------------------------

ده من احسن الديزاين بارترنس الى اتعملت للتعامل مع الداتا بيز فى الكنترولار لانه بيريحك جامد جدا من حاجات كتير و صداع



مثال بقه بالكود عشان تعرف تعمل ريبوز و تشتغل بيها بس بص الاول على المقال الجاحد ده هيعلمك اذاى تعمل ريوزاتورى جينيرك يتعمل مره واحده و متعملوش تانى بدل ما تعد تعمل ريبوزاتورى لكل جدول (كلاس انتيتى)

<https://medium.com/@chathuranga94/generic-repository-pattern-for-asp-net-core-9e3e230e20cb>

او دى جميله جدا برضوا   
<https://www.c-sharpcorner.com/article/generic-repository-pattern-in-asp-net-core/>  
  
فى مشروع ال Entity فى فولدر هنسميه Base  
أول حاجه هتعمل انترفيس اسمه IEntityBase ده هتحط فيه الكولومس الى انت عارف انها بتتكرر فى كل الجداول ذى   
InsertUserID , InsertDate , UpdateUserID,lastUpdateDate , ID  
هنعمل ال ID من نوع Guid و ده Unique Identifier جلوبال مستحيل ارقامه تتكرر و بالتالى ده ينفع يتعمل اى دى فى اى جدول و حاجه سيكيورد فاجره و بدل ماكس +1 لا هتستخدم ده Guid.NewGuid() و هو هيعملك Random ID مستحيل يتكرر

using System;

namespace Web.Entity.Base

{

public interface IEntityBase

{

Guid ID { get; set; }

DateTime? InsertDate { get; set; }

double? InsertUserID { get; set; }

DateTime? lastUpdateDate { get; set; }

double? UpdateUserID { get; set; }

}

}

ثم هنعمل الكلاس بقه الى هتبقى BaseClass كل الانتيتى تورث منها

using System;

using System.Collections.Generic;

using System.Text;

namespace Web.Entity.Base

{

public class EntityBase : IEntityBase

{

//Guid is A global Unique Identifier mapped as uniqueidentifier in SQL and nvarchar2(38) in Orical

public Guid ID { get; set; }

public double? InsertUserID { get; set; }

public double? UpdateUserID { get; set; }

public DateTime? InsertDate { get; set; } = DateTime.Now;

public DateTime? lastUpdateDate { get; set; } = DateTime.Now;

}

}

بعد كده هنعمل كلاسات ال Entity و نورث من ال EntityBase

using Web.Entity.Base;

namespace Web.Entity.Models

{

public partial class Cities : EntityBase

{

public string CityName { get; set; }

public string CityDescription { get; set; }

public string CityCountry { get; set; }

public string Population { get; set; }

}

}

using Web.Entity.Base;

namespace Web.Entity.Models

{

public partial class Persons : EntityBase

{

public string FullName { get; set; }

public string Address { get; set; }

public int Age { get; set; }

}

}

فى مشروع ال Entity فى فولدر هنسميه Repositories

بعد كده هعمل الانترفيس ثم الكلاس بتاع ال Generic Reprosatory بقه

using Web.Entity.Base;

namespace Web.Entity.Repositories

{

public interface IRepository<T> where T : EntityBase

{

IEnumerable<T> GetAll();

T GetById(Guid ID);

void Insert(T entity);

void Update(T entity);

void Delete(Guid ID);

}

}

هنعمل بقه كلاس جينيريك جميله فيها ال insert , delete , update , GetAll , GetByID

using Microsoft.EntityFrameworkCore;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using Web.Entity.Base;

using Web.Entity.Models;

namespace Web.Entity.Repositories

{

public class Repository<T> : IRepository<T> where T : EntityBase

{

protected readonly SPSDBContext context;

private DbSet<T> entities;

string errorMessage = string.Empty;

public Repository(SPSDBContext context)

{

this.context = context;

entities = context.Set<T>();

}

public IEnumerable<T> GetAll()

{

return entities.AsEnumerable();

}

public T GetById(Guid id)

{

return entities.SingleOrDefault(s => s.ID == id);

}

public void Insert(T entity)

{

if (entity == null) throw new ArgumentNullException("entity");

entities.Add(entity);

context.SaveChanges();

}

public void Update(T entity)

{

if (entity == null) throw new ArgumentNullException("entity");

context.SaveChanges();

}

public void Delete(Guid id)

{

if (id == null) throw new ArgumentNullException("entity");

T entity = entities.SingleOrDefault(s => s.ID == id);

entities.Remove(entity);

context.SaveChanges();

}

}

}

المفروض بقه انت هتروح تعمل add لل service بتاعت ال repository فى ال StartUp.cs

public void ConfigureServices(IServiceCollection services)

{

var ConnString = Configuration["ConnectionString:DefaultConnStr"];

services.AddDbContext<SPSDBContext>(options => options.UseSqlServer(ConnString));

services.AddScoped(typeof(IRepository<>), typeof(Repository<>));

services.AddControllers();

}

سطب دى الاول عشان انا هكتبها بطريقه حديثه و جميله بدلا من الى فوق دى و هتبقى حاجه عامه مش محتاجه اكتب كونيكشن سترينج و الكلام الفاضى ده فهسطب كل ده فى بروجكيت ال startUp الى هو ال Web.API

1. **EntityFrameworkCore**
2. **Microsoft.EntityFrameworkCore**
3. **Microsoft.EntityFrameworkCore.InMemory**
4. **Microsoft.EntityFrameworkCore.Relational**
5. **Microsoft.EntityFrameworkCore.Sqlite.Core**
6. **Microsoft.EntityFrameworkCore.SqlServer**
7. **Microsoft.EntityFrameworkCore.Tools**

بعد كده هكتب الفانكشن بتاعت الكونفيجريشن سيرفيس كده و اى سيرفيس بقه هتتحط فى ال Using و ده هيعلى البرفورمانس جامد لانه اول ما هيخلص شغل هيمسح الكاش على طول من الميمورى

public void ConfigureServices(IServiceCollection services)

{

var options = new DbContextOptionsBuilder<SPSDBContext>().

UseInMemoryDatabase(databaseName: "SPSDB").Options;

using (var context = new SPSDBContext(options))

{

// add service here

services.AddScoped<SPSDBContext>();

services.AddScoped(typeof(IRepository<>), typeof(Repository<>));

services.AddControllers();

}

}

المفروض تبقى انت عامل كونتيكست يا اسطى عشان تعمل كونيكشن جوه صح بقه و هيبقى شكله كده

public partial class SPSDBContext : DbContext

{

public SPSDBContext()

{ }

public SPSDBContext(DbContextOptions<SPSDBContext> options)

: base(options)

{ }

public virtual DbSet<Cities> Cities { get; set; }

public virtual DbSet<Persons> Persons { get; set; }

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

{

if (!optionsBuilder.IsConfigured)

{

optionsBuilder.UseSqlServer("Server=.;Database=SPSDB;Trusted\_Connection=True;");

}

}

protected override void OnModelCreating(ModelBuilder modelBuilder)

{

base.OnModelCreating(modelBuilder);

}

partial void OnModelCreatingPartial(ModelBuilder modelBuilder);

}

كده خلاص ياباشا تقدر تستخدم الريبوزاتورى فى الكنترولار بقه تعالى اوريك اذاى و ليكن انا عاوز اجيب كل المدن اعملها اذاى بالريبوزاتورى

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Mvc;

using Microsoft.Extensions.Logging;

using Web.Entity.Models;

using Web.Entity.Repositories;

namespace Web.API.Controllers

{

[ApiController]

[Route("[controller]")]

public class WeatherForecastController : ApiBaseController

{

private IRepository<Cities> \_Citiesepository;

public WeatherForecastController(ILogger<WeatherForecastController> logger , IRepository<Cities> Citiesepository)

:base(logger) {

\_Citiesepository = Citiesepository;

}

[HttpGet]

public IActionResult Get()

{

var Data = \_Citiesepository.GetAll().ToList();

return Ok(Data);

}

[HttpPost]

public IActionResult Post()

{

Cities item = new Cities();

item.ID = Guid.NewGuid();

item.CityName = "Cairo";

item.CityDescription = "Cairo is The Best City in Africa";

item.Population = "50,000,000 People";

\_Citiesepository.Insert(item);

return Ok(item);

}

}

}

خلى بالك انى ظبطت ال APIBaseController عشان يشتغل صح و عملته كده و شيلت منه الكونتيكست الى كنت حطه

public class ApiBaseController : ControllerBase

{

protected readonly ILogger<WeatherForecastController> \_logger;

public ApiBaseController(ILogger<WeatherForecastController> logger)

{

\_logger = logger;

}

}

كده خلاص يا برتس الريبوزاتورى اتعمل و استخدمناه كمان

طيب فى تقسيمه جميله جدا و هو انى اعمل الكنترولار لوحده و البروسيس الى جواه فى سيرفيس لوحدها

يبقى كده هعمل حاجه اسمها Services فى فولدر ال ControllerServices  
أولا هعمل كلاس بنفس اسم كلاس الكنترولار بس هكتب بدل كنترولار هكتب Service و لما اخلص كتابه الامبلمينتسشن هعمل للسرفيس دى الانترفيس بتعها ادى الكلاس فيه GetAllCities , AddCity

using Web.Entity.Repositories;

namespace Web.API.ControllersServices

{

public class WeatherForecastService : IWeatherForecastService

{

private IRepository<Cities> \_Citiesepository;

public WeatherForecastService(IRepository<Cities> Citiesepository)

{

\_Citiesepository = Citiesepository;

}

public IEnumerable<Cities> GetAllCities() => \_Citiesepository.GetAll();

public Cities AddCity([FromBody] Cities VM)

{

Cities item = new Cities();

item.ID = Guid.NewGuid();

item.CityName = VM.CityName;

item.CityCountry = VM.CityCountry;

item.CityDescription = VM.CityDescription;

item.Population = VM.Population;

\_Citiesepository.Insert(item);

return item;

}

}

}

هنا بقه هنعمل الانترفيس بتاع الكلاس دى

namespace Web.API.ControllersServices

{

public interface IWeatherForecastService

{

Cities AddCity([FromBody] Cities VM);

IEnumerable<Cities> GetAllCities();

}

}

عملت كده السيرفيس كلاس و انترفيس يبقى اروح اضيفها ضمن السيرفيس الخاصه بالبروجيكت فى ال StartUp

public void ConfigureServices(IServiceCollection services)

{

var options = new DbContextOptionsBuilder<SPSDBContext>().

UseInMemoryDatabase(databaseName: "SPSDB").Options;

using (var context = new SPSDBContext(options))

{

// add service here

services.AddScoped<SPSDBContext>();

services.AddTransient<IWeatherForecastService, WeatherForecastService>();

services.AddScoped(typeof(IRepository<>), typeof(Repository<>));

services.AddControllers();

}

}

كده خلاص اقدر استخدمها فى الكنترولار برحتى و لاحظ بقه الكنترولار صغر اذاى

namespace Web.API.Controllers

{

[ApiController]

[Route("[controller]")]

public class WeatherForecastController : ApiBaseController

{

private readonly IWeatherForecastService \_CTRL\_Service;

public WeatherForecastController(ILogger<WeatherForecastController> logger , IWeatherForecastService CTRL\_Service )

:base(logger) {

\_CTRL\_Service = CTRL\_Service;

}

[HttpGet]

public IActionResult Get()

{

var Data = \_CTRL\_Service.GetAllCities();

return Ok(Data);

}

[HttpPost]

public IActionResult Post(Cities VM)

{

Cities item = \_CTRL\_Service.AddCity(VM);

return Ok(item);

}

}

}

كده شكرا يابرنس على الريفاكتورينج الجميل ده للكود

كده بقه ناقصنا حاجه اسمها ال AutoMapper

MB

################################################################

AutoMapper

-------------------

الاوتو مابر ده يا برنس عباره عن باكيدج بتريحك فى انك تعرف تعمل مابينج بسهوله بين ال EntityClass , ViewModelClass

و دايما الى فيها الداتا هى ال Source و الى رايح اخزن فيها الداتا و اعمل معاها المابينج اسمها ال Destination فهى مثلا لو فى بروبرتى فى السورس اسمه DOB تاريخ الميلاده و عاوز اخليه يتحط فى السن Age فى الفيو و الاتنين مختلفين تماما فى الداتا تيب اهه ذى ما انت شايف دى اعملها اذاى ؟؟  
دى عشان اعملها بالكود هيجى يحصل مابنج بين الاتنين بروبرتى فال DOB مش هيعمل مابينج مع age بسبب اختلاف الاسم و اختلاف ال DataType كمان و بالتالى هيتسجل فى السن Null طيب احل المشكله دى اذاى ؟؟

بال AutoMapper و على فكره فى امثله كتير تقدر تستخدم فيها الاوتومابينج ذى ربط الجداول و بعضها و حوارات كبيره كده هتعرفها كل شويه عنه و عشان كده اقرأ الدوكمنتيشن ده جميل جدا عن ال AutoMapper  
<https://docs.automapper.org/en/latest/Getting-started.html>  
و ده مقال يفهمك بسرعه   
<https://stackoverflow.com/questions/40275195/how-to-set-up-automapper-in-asp-net-core>  
و ده مقال يفهمك احسن برضوا   
<https://www.c-sharpcorner.com/article/integrate-automapper-in-net-core-web-api2/>  
  
و تعالى بقه لمثال كويس اهه   
اولا لازم تسطب الاتنين باكيدجيس دول فى ال Web.API فى المشروع الى انت هتكتب فيه كود :D

Install-Package AutoMapper

Install-Package AutoMapper.Extensions.Microsoft.DependencyInjection

بعد كده هروح على ال StartUp.cs و احط فى الكونفيجريشن السطر ده

public void ConfigureServices(IServiceCollection services)

{

var options = new DbContextOptionsBuilder<SPSDBContext>().

UseInMemoryDatabase(databaseName: "SPSDB").Options;

using (var context = new SPSDBContext(options))

{

// add service here

services.AddScoped<SPSDBContext>();

services.AddTransient<IWeatherForecastService, WeatherForecastService>();

services.AddScoped(typeof(IRepository<>), typeof(Repository<>));

services.AddAutoMapper(AppDomain.CurrentDomain.GetAssemblies());

services.AddControllers();

}

}

بعد كده هعمل اتنين كلاس واحد Entity و هو موجود فعلا و هيبقى بتاع ال Cities الى موجود فى الانتيتى و عامل مابينج مع الداتا بيز

namespace Web.Entity.Models

{

public partial class Cities : EntityBase

{

public string CityName { get; set; }

public string CityDescription { get; set; }

public string CityCountry { get; set; }

public string Population { get; set; }

}

}

و كلاس ال ViewModel ده و ذى ما انت شايف انى شيلت كلمه City من كل البروبرتيس و ده المفروض هيخلى المابينج ميحصلش الا لو عملت كونفيجريشن بقه بالاوتومابر يا صديقى او اعمله بالكود بقه و يبقى حاجه مقرفه فى التعديل لو هعمله فى كذا مكان و احاول اعدل عليه بعد كده

namespace Web.API.ViewModels

{

public class CitiesVM : EntityBase

{

public string Name { get; set; }

public string Description { get; set; }

public string Country { get; set; }

public string Population { get; set; }

}

}

هنيجى فى الكنترولار نخلي الداتا تتحط كده فى ال Cities مره و مره فى ال CitiesVM

[HttpGet]

public IActionResult Get()

{

Cities Data = \_CTRL\_Service.GetAllCities().FirstOrDefault();

return Ok(Data);

}

{

    "cityName": "Cairo",

    "cityDescription": "Cairo is The Best City in Africa",

    "cityCountry": Egypt,

    "population": "50,000,000 People",

    "id": "dfc143db-f8b0-4573-ae62-1f03a49c515b",

    "insertUserID": null,

    "updateUserID": null,

    "insertDate": "2020-07-25T17:27:35.3800284",

    "lastUpdateDate": "2020-07-25T17:27:35.3800344"

}

طيب انا دلوقتى لو حابب احط حاجه من نوع ال Cities فى حاجه من نوع CitiesVM ده اصلا مينفعش الا عن طريق المابينج و بالتالى اول حاجه هنعملها اننا نعمل بقه المابينج بين الاتنين كلاس عشان اعرف احط الى فى كلاس Source الى هو Cities فى كلاس الديستينيشن الى هو ال CitiesVM  
فهروح رايح عامل فولدر اسميه AutoMapper و احط جواه كلاس ال AutoMapperProfile الى بيورث من ال كلاس ال Profile

و الى جواه الكلاس ده بحط فيه كل المابينج الى انا عاوز اعملها و ممكن تعمل كذا كلاس لو حابب تقسم كل اوتومابر بروفايل لوحده

عمتا انا هنا هعمل جوه الكونستراكتور كريت ماب و اقوله حطلى الداتا الى فى ال Cities جوه ال CitiesVM من غير اى كونفيجريشنز تانيه خالص فهيحصل مابينج فعلا بس البروبرتيس المختلفه مش هتنقل داتا و مش هيحصل بينهم مابينج اصلا و فقط هيحصل للمتشابهين بس

namespace Web.API.AutoMapper

{

public class AutoMapperProfile : Profile

{

public AutoMapperProfile()

{

// Source to Destination

CreateMap <Cities, CitiesVM>();

}

}

}

كده هنروح على الكنترولار بقه عشان نستعمل ال Mapping بعد ما عملناه و هيبقى بالشكل ده

namespace Web.API.Controllers

{

[ApiController]

[Route("[controller]")]

public class WeatherForecastController : ApiBaseController

{

private readonly IWeatherForecastService \_CTRL\_Service;

private readonly IMapper \_mapper;

public WeatherForecastController(ILogger<WeatherForecastController> logger ,

IWeatherForecastService CTRL\_Service , IMapper mapper)

:base(logger) {

\_CTRL\_Service = CTRL\_Service;

\_mapper = mapper;

}

[HttpGet]

public IActionResult Get()

{

Cities Data = \_CTRL\_Service.GetAllCities().FirstOrDefault();

CitiesVM Item = \_mapper.Map<CitiesVM>(Data);

return Ok(Item);

}

[HttpPost]

public IActionResult Post(Cities VM)

{

Cities item = \_CTRL\_Service.AddCity(VM);

return Ok(item);

}

}

}

ذى ما انت شايف مينفعش احط ال Cities فى ال CitiesVM بس بال AutoMapper خلاها تنفع عافيه بس للاسف انا معملتش كونفيجريشنز فالى هيحصل ياباشا ان الاعمده الى اسميها مختلفه مش هيجى فيها داتا و هتبقى النتيجه كده

{

    "name": null,

    "description": null,

    "country": null,

    "population": "50,000,000 People",

    "id": "dfc143db-f8b0-4573-ae62-1f03a49c515b",

    "insertUserID": null,

    "updateUserID": null,

    "insertDate": "2020-07-25T17:27:35.3800284",

    "lastUpdateDate": "2020-07-25T17:27:35.3800344"

}

طيب انا عاوز اقوله ال Cityname , Citydescription , Citycountry فى ال Cities class هى هى

name , description , country فى ال CitiesVM class   
  
ده ياباشا يحصل بكل سهوله فى ال AutoMapper Profile و انت بتعمل Creation للمابينج اهه  
هنا بقوله بص اعملى ماب بين السورس Cities الى فيها الداتا و هنقلها للديستينيشن الى هى هتاخد الداتا الى هى CitiesVM و بقوله بقه لكل عضو معين العمود الى اسمه Name فى الديستينيشن روح خد الداتا بتعته من السورس الى اسمه CityName وهكذا بقه فى الى انا عاوزه ذى ما انت شايف

namespace Web.API.AutoMapper

{

public class AutoMapperProfile : Profile

{

public AutoMapperProfile()

{

//createMap from Source to Destination

// but ForMember from Destination to Source

CreateMap<Cities, CitiesVM>()

.ForMember(dest => dest.Name, opt => opt.MapFrom(src => src.CityName))

.ForMember(dest => dest.Country, opt => opt.MapFrom(src => src.CityCountry))

.ForMember(dest => dest.Description, opt => opt.MapFrom(src => src.CityDescription));

}

}

}

فهتكون النتيجه كده و ذى ما انت شايف بقه كانى عامل الداتا بالكلاس الاصلى و ده كله بسبب استخدام ال AutoMapper

{

    "name": "Cairo",

    "description": "Cairo is The Best City in Africa",

    "country": "Egypt",

    "population": "50,000,000 People",

    "id": "dfc143db-f8b0-4573-ae62-1f03a49c515b",

    "insertUserID": null,

    "updateUserID": null,

    "insertDate": "2020-07-25T17:27:35.3800284",

    "lastUpdateDate": "2020-07-25T17:27:35.3800344"

}

و على فكره الاوتومابر ليه استخدامات كتير جدا جدا هتعرفها من الموقع الرسمى بتاعه و ياريت تبقى تقرأ عن ال Nested Mapping من هنا   
  
<https://www.c-sharpcorner.com/article/integrate-automapper-in-net-core-web-api2/>

و الى فيه بتقدر تعمل مابينج بين كولوم و أوبجيكت و ده بيساعد جدا جدا فى ال Relations بتاعت الداتا بيز بيسهل الشغل فيها جدا جدا   
ابقى بص على المثال الى قولتلك عليه هتحس ان الدنيا بقه فله شمعه منوره

و بالمره على حوار ال Conditional Mapping

MB

################################################################

AutoMapper

-------------------