C# ASP.NET Core 3.0

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

MB

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

Core Intro

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

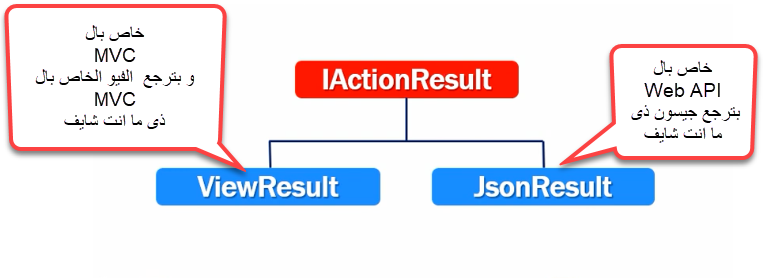
بص يا سيدى دى تقنيه جديد ميكروسوفت عملتها عشان تريحنا من الاخر و خلت ان التقنيه دى تجمع بين اهم 3 حاجات ممكن تحتاجها فى اى فريم ورك و هى انها خليتها   
Open Source : و ده معنله ان اى حد ممكن يخش يعدل فيها و يطور فيها و يبعت ليهم التصليحه و ده بيتيج للناس انها تعمل حاجات كتير مفيده جدا للكور

Cross platform : و ده معناه ان البروداكشن بتعها هيشتغل على windows , Mac , linux اي حاجه و بالتالى دى كده فله شمعه منوره

Performance : اهتموا بانها تبقى تقنيه سريعه جدا لدرجه ان فى ناس بتقول انها اسرع حاجه موجوده حتى الان فى الكوكب :D

ايه طيب اهميه ال CORE او هتفيدنى فى ايه ؟؟

1. Cross platform يعنى اقدر احطه على windows , mac , linux و كمان اقدر اعمله هوست على   
   IIS , Apache ,Docker اى حاجه عادى خالص مش لازم IIS ذى باقى تقنيات ميكروسوفت و بالتالى كده انا ممكن ارفع المشروع على Apache بهوست رخيص بدل هوستات ميركوسوفت المكلفه جدا
2. سهله جدا فى التيست بس لازم تعرف بقه شغل ال unit testing هنشرحه بعدين ده بعدين خالص ههههه
3. احسن حاجه فيها انها بتوفر ستراكتشار واحد لكل من ال web api , MVC يعنى تقدر تشتغل الاتنين فى مشروع واحد الى هو اشتغل الى يريحك و يخدم مشروعك ياباشا احنا مش هنعطلك و ده لان الكنترولر بتاع ال MVC او Web API كلهم بيورثوا من نفس ال Controller base Class و كلهم جواهم الميثودز بترجع IActionResults و دى كلاس معدل احسن من ال HttpRequest , HttpResponse و ميزته انه عباره عن انترفيس لو بتتعامل مع ال MVC هيرجعلك ViewResult و لو بتتعامل مع web API هيرجعلك Json Result و بكده هو جمع بين انه ينفع يشتغل الاتنين فى نفس الكنترولر عادى خالص



1. بتدعم حته ال modularity الى هو ان يبقى فى Middleware ينظموا الشغل و كده و الكلام فى الموضوع ده هيبقى بالتفصيل بعدين
2. بتدعم ال dependency injection عن طريق فريمورك اسمه IOC container و دى انك تعمل سيرفيس معينه و لما تعوز تكلمها بسهوله بتروح رايح على الكونستراكتور بتاعك مستدعيها فيه و بعد كده تعايش معاها و اتعامل و هيتم شرحها بالتفصيل بعدين لا تقلق و من اهم الحاجات الى انا بستخدمها فى الموضوع ده هو عمل Reposotries اهميتها بقه ان انت مش بتعمل اوبجيكت من الكلاس عشان تستدعيه و الى كان بيسبب مشاكل لو انت غيرت الكونستراكتور بتاع الكلاس الى انت بتعمل منه اوبجيكت و خلاك تعمل الانجيكشن و تنسى الكلاس الى بتعمل منه انيستانس للابد

بمعنى اصح المشكله كانت ان انا كنت لما اجى استخدم استخدم اوبجيكت من كلاس ما فكنت بقوله

className obj = new ClassName()

فكان الكونستراكتور كده فاضى و شغال عادى و شغال فى المشروع كله و استخدمته مليون مره

و جه مره كده من المرات عزنا نبعت براميتار من الكونستراكتور ده فكان الموضوع ده مشكله بنت كلب لانك كده انت مضطر تلف على المشروع كله تغيير الدفنيشن و تعمله كده

className obj = new ClassName(param)

فمن هنا ظهر الحل و هو ال Dependency injection

و ده اهميته ان خلاص بقىت تروح على الكلاس الى انت هتستخدم فيه الاوبجيكت من الكلاس بتاعك و تحطه كبراميتار فى الكونستراكتور بتاعه و كده انت عملت ال dependenct injection و تقدر تستخدم انستانس من الكلاس بتاعك ده براحتك و لو غيرت فى الكونستراكتور بتاعه و حطيت 10000 براميتار مش هيعمل اى مشكله ولا هيقولك بتعمل ايه و هيشتغل عادى

شكله بيبقى كده

using System;

namespace MyConsoleApp

{

class Program

{

static void Main(string[] args)

{

MyClass MyClass = new MyClass(0, 0);

cls x = new cls(MyClass);

}

}

class cls

{

**private MyClass MyClass;**

**// Dependency injection**

**public cls(MyClass MyClassInj)**

**{**

**this.MyClass = MyClassInj;**

**this.MyClass.add(5, 6);**

**}**

}

class MyClass

{

public MyClass(int x, int y)

{

Console.WriteLine("Hello MyClass");

}

public void add(int x, int y)

{

Console.WriteLine(x+y);

}

}

}

ملحوظه هامه ال Dependency injection بيعتمد فى شغله على ال singleton pattern و ده هو انى اذاى اعرف اخد اوبجيكت واحد بس من الكلاس بمعنى انى انا ممكن فى كلاس ما اروح اعمل 10 اوبجيكت من كلاس اخر و استخدمهم و ده حاجه وحشه جدا للميمورى فجه السجنيلتون باترن قالك لأ انا هخلى انك فى الكلاس الواحد تقدر تستخدم اوبجكت واحد بس من الكلاس التانى ده و لو فكرت تعمل اوبجكت تانى فهو هيعتبره مش موجود و هيبقى كل الشغل على الاوبجيكت الاولانى و كاننا عاملين شير للاوبجيكت الاول

الاوبجكت فى السجنيلتون بيبقى عامل ذى الابن الوحيد لابوه :D و ابوه ميت يعنى امه مش هتجيب غيره

الابن هنا هو الاوبجيكت

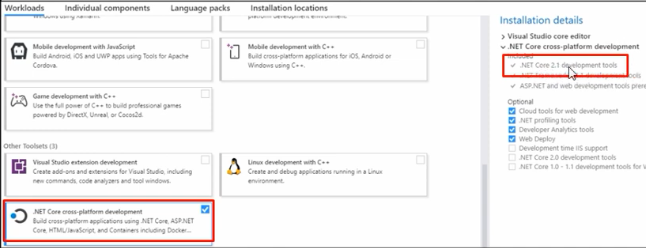
الاب هنا هو الكلاس الى انا بعمل منه الاوبجيكت

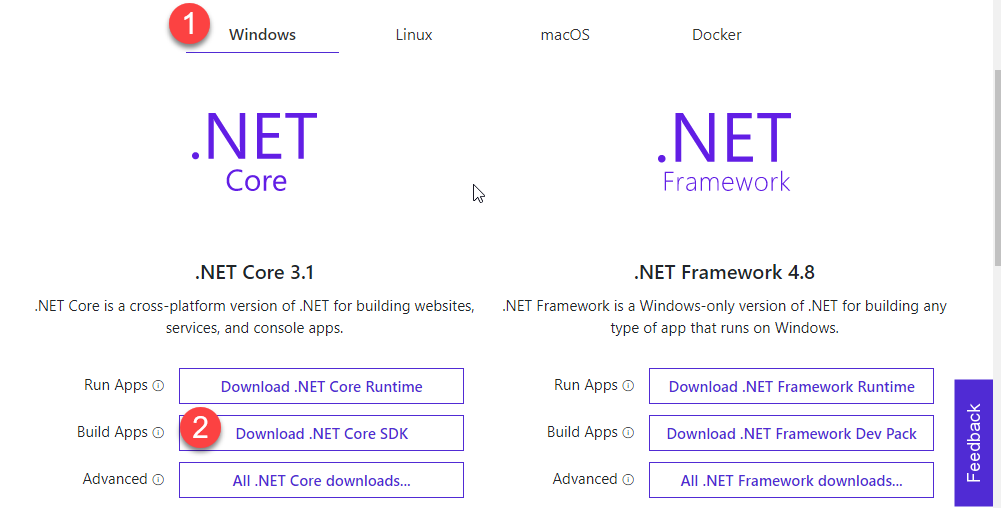
الام هنا هى الكلاس الى انا بروح اعمل الاوبجيكت الابن جواه و بما انها خلفت واحد و متجوزتش تانى فمش و شالت الرحم بالسجنيلتون فمش هتعرف تخلف تانى و تجيب عيال تانى و بكده مش هتعرف تعمل اوبجيكت جواها من نفس الاب تانى لان الاب الله يرحمه و ساب عيل وحيد حد فاهم حاجه ههههههههههههههههههههه

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

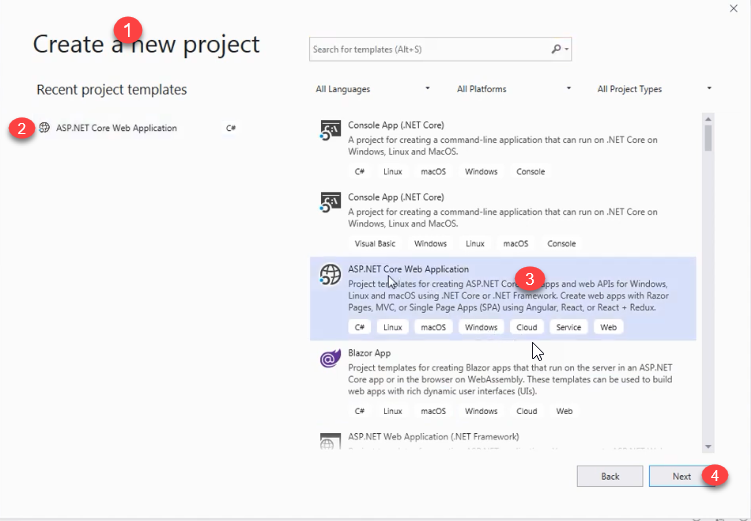
متطلبات الكورس من حيث الادوات

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

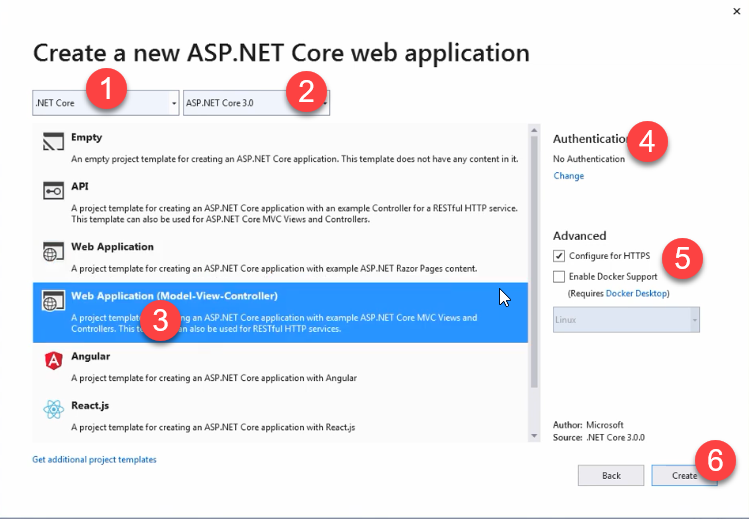
طبعا احسن حاجه تحملوا Visual Studio 2019 و انتوا بتسطبوه بقه تختاروا الاختيارات دى عشان الكور يشتغل معاكوا و يبقى كده انتوا حملتوا ال Net CORE SDK من خلال التسطيب يا رجاله   


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>  


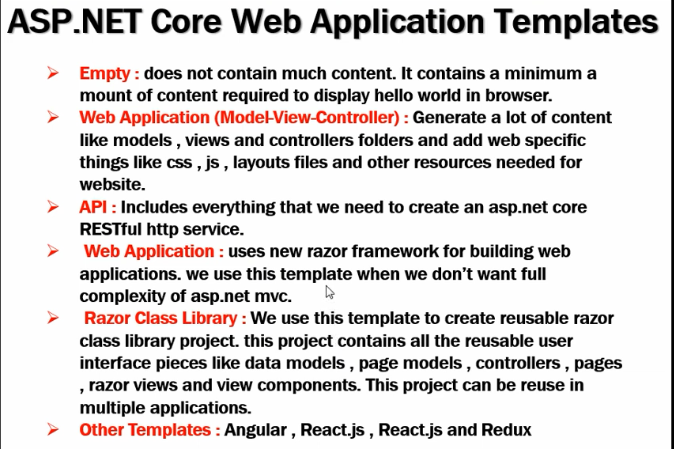
و لما اجى اكريت بقه بروجيكت كور اتبع الكلام ده



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



ذى ما انت شايف اخترت انه يبقى كور و و انه يبقى Asp.net core 3 عشان ده احدث حاجه دلوقتى و اخترت انى هشتغل المشروع ده MVC و قولتله مفيش Auth جاهزه لا انا الى هعمل ال Auth بنفسى custom من الاخر و فى غالب الامر بيتم اختيار individual Auth عشان يبقى عندك حاجه Built in تعتمد عليها برضوا لو عصلجت معاك او لو كان الحوار مش محتاج تعقد و ينفع معاه الشغل العادى الى هو بيبقى جاهز عادتا كده :D خليته يدعم ال HTTPS و بعد كده بقوله اعملى المشروع ياعم بقه ودى بقه شويه شرح بسيط لشويه التيمبليتش الى فوق دول



احنا فى الكورس هيبقى تركيزنا كله على MVC , API , Wep application

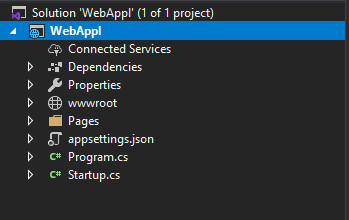
و عشان تفرق صحيح بين ال MVC , API ال MVC هتلاقى فيه Model – view – Controllers فى حين ان ال API بيبقى فيه فقط Controllers و ممكن تعمله Model بس مستحيل تعمله View عشان اصلا ده بيبقى بيتكلم ب json مع مشاريع تانيه او فرونت ايند فبيبقى مش محتاج ال View خالص

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

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 WepAppl هيفتح معايا WEpAppl.csproj ملف المشروع من الاخر و ده بيبقى فيه كل فولدرات البروجيكت و الريفرنس و الباكيدجات الى انا حملتها

**Dependencies** دى ذى ال References و دى هنا متقسمه ل 3 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

Pages : دى تحتوى على صفحات ال Razor الى المفروض تبقى هى الفرونت ايند بقه

**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 WebAppl

{

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 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**

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**

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

Razor pages

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

دى بقه يا سيدى تقنيه اتعملت عشان تعمل ال HTML بال CSharp فهتلاقى ان كل فايل HTML صيغته .cshtml

و ده بيبقى بينقسم لملفين فعلى سبيل المثال اول ملف فى المشروع بيبتدى هو ال index.cshtml هتلاقى جواه ملفين

الملف الاول هو index.cshtml و الملف ده بيحتوى على ال HTML و على Razor calls

الملفت التانى هو index.cshtml.cs و ده بنسميه بشكل عام ال Page Model و ده بقه بكتب جواه ال Csharp الى اقدر بيها اروح للملف الاول و اعمل databinding و اخد الداتا الى فى ال Csharp و احطها فى الملف ال HTML

و ده مثال بسيط يوضح الامور

Index.cshtml.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.RazorPages;

using Microsoft.Extensions.Logging;

namespace WebAppl.Pages

{

public class IndexModel : **PageModel**

{

private readonly ILogger<IndexModel> \_logger;

public IndexModel(ILogger<IndexModel> logger)

{

\_logger = logger;

}

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

// onGet like Get in http request

public void OnGet()

{

**this.Name = "Mahmoud Badawy";**

}

}

}

فى المثال ده انا عملت property اسمها Name و جوه ال OnGet قولتله حط فيها اسمى و شكرا على كده المفروض بقه اروح لفايل ال HTML و اقوله ياباشا عندك فى المودل بروبارتى اسمه Name هاتهولى

طب الكلام الى فوق ده يتعمل اذاى ؟؟

داخل اى HTML بروحش اقوله   
@Model.PropertyName

يعنى كانه بقوله فى الموديل بتاعك خش هتلاقى بروبارتى اسمه فلان الفلانى هاتلى قيمته

و ادى ال index.cshtml

@page

@model IndexModel

@{

ViewData["Title"] = "Home page";

}

<div class="text-center">

<h1 class="display-4">Welcome</h1>

**<p> @Model.Name</p>**

</div>

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

MVP – MVC - MVVM

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

MVP : Model – view – presenter

MVC : Model – view – Controller

الفرق بينهم ايه بقه ؟؟

بص يا سيدى ال MVP ده هو الباترن الى كانت ال web forms شغاله بيه و هو ان كان فيه presenter ده هو الكلاس الى فيه Events بتاع كل صافحه و دى كانت بتربط بين الداتا بيز و الصفحه الخاصه بالكلاس ده و بس و مينفعش استخدم الكلاس بتاع ال presenter ده فى اى مكان تانى و لو حتى فى فانكشنالتى هتتكرر فى صفحه تانيه كان لازم تكررها و مينفعش تستخدم المعموله افتكر شغلك فى الويب فورمس و انت هتفهم يا باشا الى بقوله

و طبعا كده البترن ده خره اوى يعنى

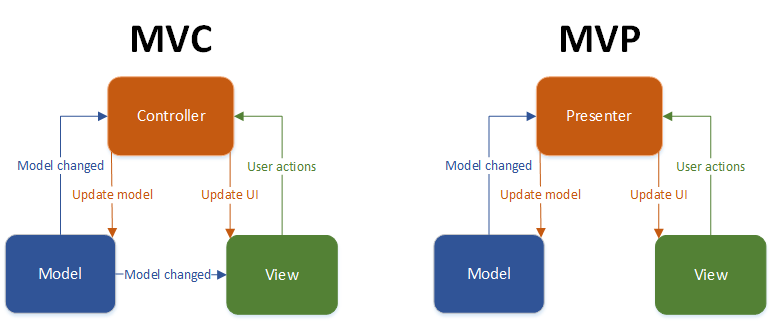
من الاخر index.aspx ده الفيو و index.aspx.cs ده ال presenter و اى كلاس بقه خاص بالداتا بيز هو ال Model

فقاموا عاملوا ايه بقه عشان يحلوا مشكله تكرار الكود الكتير دى ؟؟

قالك احنا نعمل حاجه اسمها MVC الى هيبقى فيها الداتا بيز الى فى كلاسات المودل بتكلم حاجه اسمها كلاسات الكنترول الى هي فيها البيزنيس بتاع المشروع او اللوجيك و كلاسات الكنترول دى بتكلم الفيو الى هيظهر الكلام فى الصفحه

و الميزه فى ال MVC بقه ان اى Model ينفع يكلم اى controller و اى controller ينفع يكلم اى View عادى خالص فكده احنا مش مضطرين نكرر الكود فى اماكن كتير ذى الباترن الزفت الى فاتت

و دى صوره توضح الكلام الى بقوله بسرعه



لقوا ان مثلا انا ممكن عندى صفحات ذى صفحه ال login بكتب كودها مره واحده و كذلك صفحات ال

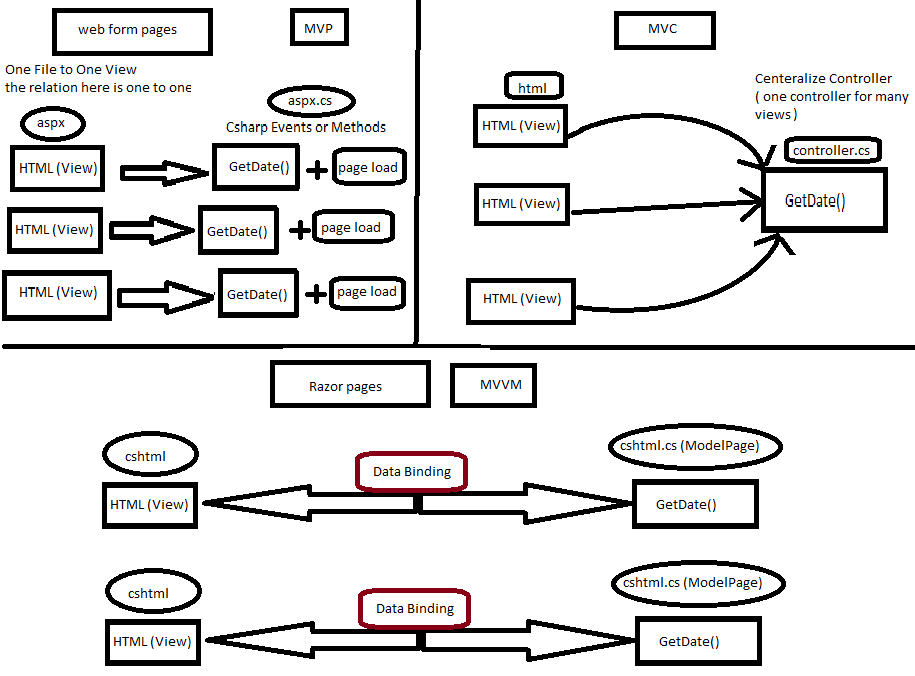
About us , Contact us فقالك يا سيدى طب ما بدل ما استخدم ال MVC بس منا استخدم الاتنين مع بعض و اسهل على نفسى

يعنى فى الصفحات الى الكود بتاعها مش هيتكرر استخدم طريقه ال MVP عشان دى سهله عباره عن صقحه و ليها الايفناتات بتعتها لما ادوس على زرار تروح تنفذ من الفايل الى جبنها و خلاص ذى ما بيحصل فى الويب فورم

و فى حاله ان فى فانكشن بقه هتتكرر كتير جدا فى المشروع كله ذى مثلا انى اجيب داتا معينه محتاجه فى صفحات و اماكن كتير فى الموقع و متوقع انى هناديلها خلال المشروع 100 مره فى 100 صفحه فكده مينفعش فكره الفايل الى جنبه ده بتاعت ال MVP خالص و الويب فورمس دى و بكده دى انا اعملها بشكل مركزى فى كنترولار معين عشان اقدر اكلم الكنترولار ده من اى view الى هما 100 صفحه و بكده انا خليت 100 صفحه يكلموا فايل واحد ووفرت على السيرفر 100 صفحه يعنى مساحه اقل و سرعه اكبر للمشروع و كمان مخلتش المشروع يعمل لودينج كتير كل شويه لايفينتات ملهاش اى لازمه ذى ما كان بيحصل فى ال MVP ايام الويب فورمس دى كانت خره كنت تروح تدوس على زرار اجرب تلاقى الصفحه كلها حملت من جديد عشان حوار ال postback المقرف ده

ال MVVM ده بقه اتعمل لصفحات ال Razor عشان يدعم ال databinding و ده ذى الويب فورمس بالظبط بس الفرق ان هنا بيحصل داتا بيندينج و بالتالى قلل من حوار ان كل ايفينت فى الويب فورمس يبعت ريكويستات ملفهاش لازمه للسيرفر يعنى من الاخر الفيو الى هو ال cshtml بيكلم ال ModelPage الى هى cshtml.cs بتعته و يروح يجيب منها بيانات على حسب الى الكلاينت طالبه بس و مبيجبش حاجه زياده ولا بيشغل اى فانكشن او ايفنت مش مطلوبه ذى ما كان بيحصل فى الويب فورمس فبيريح السيرفر كتير جدا جدا

فهتلاقى ان ال razor عامل ذى حوار الكومبونانت بتاع الانجيولار بالظبط فى حته ال html و ts هو نفس الفكره يعتبر حتى الاتنين فيهم داتا بيندينج ياعم الحج واضح ان ميكروسوفت مش هتبطر تكبير شيتات من صحابها



و بما ان Razor تحديث جميل حصل و بقى ذى الويب فورمس بس متطور جدا يبقى الاحسن انى اعمل الصفحات الى ذى ال login , contact us , about us بيها و فاكس من الويب فورمس بقه خالص

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

Razor Pages

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

أول حاجه روح اعمل برجيكت من نوع ASP Core Web Application هتقلافى المشروع اتهمل فهتروح على index و اول ما تشوف صيغه cshtml و cshtml.cs تعرف على طول ان ده razor و ده لان ال Razor معمول اساسا عشان تعرف تكتب ال C# جوه ال HTML تعالى بقه على فايل ال index.cshtml الى ببقى كاتب فيه ال HTML و هنتعرف على syntax ال Razor بقه

بص اول حاجه

@page دى ديركتيف كده لازم يبقى موجود فى أول سطر فى اى صفحه رازور

@indexModel دى معناها ان الصفحه دى تروح تجيب كود ال Csharp بتعها من الكلاس الى اسمه indexModel يعنى بتشاور على الكلاس بتاع صفحه ال Index من الاخر

عشان تكتب كود سى شارب جوه ال html هنا بقه بتبقى الطريقه كده

@{ your C# code + Html if you need }

@{

string MyFunc()

{

var FriendName = "Karim Salah";

return FriendName;

}

<p> My friend is : @MyFunc() </p>

}

@{

var Son = "Mostafa";

var age = 25;

@:My son is : @(Son + " " + this.Model.Name)

<br />

<span>MyFunc Age is : </span> @age

}

انت بتعمل @ و بعد كده بتفتح قوسين مجموعه و جواهم بقه بتكتب كود سى شارب

و لو كتبت اى حاجه HTML بتتقرأ عادى بس لو هتنادى جوه تاجات ال HTML اى حاجه Csharp بتروح حاطط قبلها @ عشان تعملها calling و تشتغل ذى ما انت شايف لما جيت اعمل call للفناكشن فى تاج ال paragraph

و لو عاوز اكتب تيكست عادى جوه السى شارب بكتبه @:MyText و ده لان بال @ و بعدها نقطتين كانى بقوله اوقف تشغيل السى شارب فى الجمله الى جايه دى

و عندم بقه لو كتبت @ و فتحت قوسين () ده كده بيتيح ليا انى اعمل جوه القوس ده اى عمليه ذى جمع طرح او concat

بالنسبه ل this.Model.Name دى يا سيدى فاريبول انا جايبها من Csharp code الى مكتوب فى ال PageModel الى هو ال index.cshtml.cs

و دى بقه طريقه نداء كل حاجه لوحدها الى هى طريقه الفرط هههههههه

<p> @this.Model.Name</p>

<p>Today is : @DateTime.Now</p>

<p> last time of playing PS4 was at : @(DateTime.Now - TimeSpan.FromDays(7))</p>

<p> The Call of @@MyFunc() is Return : @MyFunc() </p>

الكومنتات هنا بتتكتب كده

@\* this is A comment on Razor \*@

الفورم لوب و السويتش و الايف كونديشن و ال while و foreach بيتكتبوا كده الى هو @ قبل ال key الخاصه بيهم و هو بيفهم بقه باقى الاقواس بتعتهم

<div>

@for (var x=0;x<10;x++)

{

<p> Number : @x </p>

}

</div>

كمان انت تقدر تكتب الفانكشنز كلها فى مكان مخصص للفانكشنز بس و تناديها بقه فى اى حته فى الصفحه و بتتكتب بالطريقه دى

@functions{

public string MyFunc()

{

var FriendName = "Karim Salah";

return FriendName;

}

}

<p> My friend is : @MyFunc() </p>

**ذى ما انت شايف عملت فانكشن جوه الرازور فانكشنز و ناديتها بعد ما قفلت قوس الفانكشنز**

Index.cshtml

@page

@model IndexModel

@{

ViewData["Title"] = "Home page";

}

<div class="text-center">

<h1 class="display-4">Welcome</h1>

<p> @this.Model.Name</p>

<p>Today is : @DateTime.Now</p>

<p> last time of playing PS4 was at : @(DateTime.Now - TimeSpan.FromDays(7))</p>

<p> The Call of @@MyFunc() is Return : @MyFunc() </p>

@\*

this is A comment on Razor

\*@

@{

string MyFunc()

{

var FriendName = "Karim Salah";

return FriendName;

}

<p> My friend is : @MyFunc() </p>

}

@{

var Son = "Mostafa";

var age = 25;

@:My son is : @(Son + " " + this.Model.Name)

<br />

<span>MyFunc Age is : </span> @age

}

<div>

@if (Son == "Mostafa")

{

<p> Hello : @Son</p>

}

else

{

<p> Hi : @Son</p>

}

</div>

<div>

@for (var x = 0; x < 10; x++)

{

<p> Number : @x </p>

}

</div>

</div>

طبعا كل الى فات انى اكتب لوجيك و اكتب كود كتير فى ال HTML ده كده حاجه خره خره يعنى و متنفعش بنفعه و فكك منها كل الى فات ده بلح زغلول ملوش لازمه ده عشان تبقى فاهم قوه اللغه مش اكتر تعالى بقى للصح يا كبير و هو انى بكتب الكود فى ال PageModel الى هو مكان كتابه ال CS و انادى بس الحاجات الى انا عوزها فى ال HTML و اعمل DataBinding بين ال HTML , Csharp الى هو تطبيق مفهوم ال MVVM

Model view view Model

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

Razor from Cs (Best Practise)

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

عاوز بقه اوريك اذاى بيحصل داتا بيندينج بين فايل السى شارب و فايل ال Html لان ده هو اصلا الى انت هتشتغل بيه على طول و فاكس الكلام العبيط الى فوق ده

يعنى من الاخر المفروض هتكتب كود السى شارب فى ملف ال Cs و كود ال Html فى ملف ال HTML و اى حاجه هتحتاجها من ال CS هتناديها فى سطر صغنن و خلاص النداء الفرط ده ههههههههههه

تعالى اوريك انى حولت الكلام العك الى فوق للصح اذاى بقه مع تغييرات طفيفه يعنى مسحت الحاجات الى ملهاش لازمه و عدلت شويه حاجات كده على مزاجى و خلاص هههههههههه

ده فايل ال CS

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.RazorPages;

using Microsoft.Extensions.Logging;

namespace WebAppl.Pages

{

public class IndexModel : PageModel

{

private readonly ILogger<IndexModel> \_logger;

public IndexModel(ILogger<IndexModel> logger)

{

\_logger = logger;

}

public string Name { get; set; }

public string Son { get; set; }

public DateTime LastDate { get; set; }

public int[] Counter = new int[10]; /\* n is an array of 10 integers \*/

public void OnGet()

{

/\* initialize elements of array n \*/

for (int i = 0; i < 10; i++)

{

this.Counter[i] = i + 100;

}

this.Name = "Mahmoud Badawy";

this.LastDate = DateTime.Now - TimeSpan.FromDays(7);

if (Son == "Mostafa")

{

this.Son = "Mostafa";

}

else

{

this.Son = "No Name";

}

}

public string MyFunc()

{

var FriendName = "Karim Salah";

return FriendName;

}

}

}

بتكتب فيه كل ال Member كانه كلاس عادى خالص و مفيهوش اى مشاكل يعنى بتكتب جوه الكلاس

Properties – Methods – Variables و اى حاجه عادى بتتكتب فى كلاس يعنى

و فى ال HTML فقط بتنادى على نتائج الفاريبول او نداء الفانكشنز عن طريقه انك بتقوله @this.Model.NameOfWhatYouNeedToCall

@page

@model IndexModel

@{

ViewData["Title"] = "Home page";

}

<div class="text-center">

<h1 class="display-4">Welcome</h1>

<p> @this.Model.Name</p>

<p> last time of playing PS4 was at : @this.Model.LastDate</p>

<p> The Call of MyFunc() is Return : @this.Model.MyFunc() </p>

<div>

< p > My Son is : @this.Model.Son </ p >

</div>

<div>

@foreach (var item in this.Model.Counter)

{

< p > Number : @item </ p >

}

</div>

</div>

الموضوع بسيط خالص و سهل جدا جدا جدا

كده احنا ربطنا بين ال CS , HTML و عملنا داتا بيندينج ياباشا لان بقى الفاريبولز الى فى ال CS بيجصلها ريندرينج فى ال HTML يا كبير بسهوله

و من هنا نقدر نفهم بقه ان كده خلاص انت تقدر تعمل كلاسيس بقه منفصله و تكلمها بال CS File و تتعايش كمشورع كبير بقه و تقدر حتى تطبق دلوقتى ال MVC مع ال MVVM بتاعك هههههه ما هى ظاطت خلاص

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

Razor (Handler Mehtod)

OnGet – OnPost – OnDelete – OnPut

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

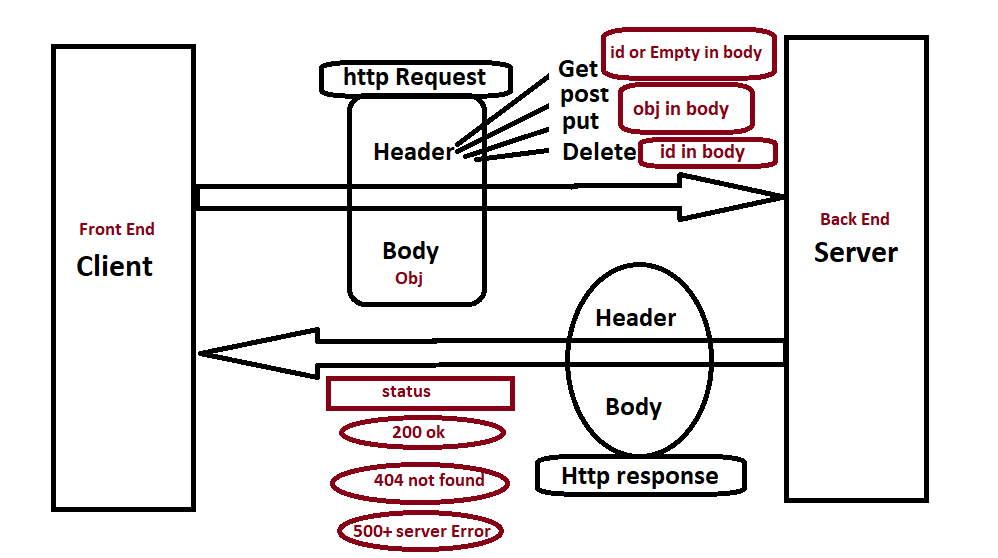
بص يا سيدى عشان تفهم الكلام ده المفروض تفهم ال Http Request و بيتم اذاى ؟؟

الى بيتم ان انت بتكلم الباك اند عن طريق ال Http Request و كانك بتبعت طلب من الفرونت ايند للباك اند و مستنى الرد فمره تبعت طلب تقولهم get الى هى هاتولى داتا اشوفها و مره تانيه تقولهم Post الى هى ضيفولى الداتا دى على الداتا بيز يا جدعان و مره تقولهم Put الى حدثولى البيانات و مره تقولهم Delete الى هى امسحولى الداتا الفلانيه

طيب عشان السيرفر يفهم انهى حاجه ينفذها اكيد فى حاجه بتدل على الكلام ده و هى ان و انت بتعمل الريكويست بتبعت حاجات فى ال Header و حاجات فى ال Body بتاع ال Request فالهيدر بقه بيشيل نوع الفانشكن الى انت محتاجها تتنفذ سواء جيت او بوست او دليت او بوت و كمان بيحتوى على ال Token احيانا و بيحتوى على اى معلومات مهمه السيرفر هيحتجها عشان يفهم الداتا و نوع الداتا الى هتوصله Json ولا XML و انت عاوز تعمل فيها ايه

و فى ال Body بتاع الريكويست انت بتبعت فيه الى انت محتاجه بقه يعنى لو هتمسح بتبعت ال ID بتاع العنصر الى هتمسحه و لو هتضيف بتبعت الاوبجيكت الى هتضيفه فى الداتا بيز و لو هتعدل بتبعت برضوا الاوبجيكت الى هتعدل عليه و لو عاوز تبين داتا معينه فعند طريفتين الطريقه الاولى انك عاوز تجيب الداتا كلها فبتعتش حاجه اما لو عاوز تعمل فلتر على حاجات معينه بتبعت الحاجات الى انت هتفلتر بيها بقه ساعتها يعنى و لنفترض عاوز تبين العنصر الى ال ID بتاعه 5 فبتبعت فى ال Body ال ID = 5 و هكذا

و دى رسمه ملخصه الكلام ده كله



طيب بعد ده كله المفروض كده انا هبعت ريكويستات من الفرونت الى هو فايل ال HTML للباك اند الى هو فايل ال CS يعنى من ال index.cshtml الى index.cshtml.cs و الرابط بينهم اسم الكلاس فى السى اس و اسم الموديل فى الاتش تى ام ال

بالنسبه بقه للسى شارب انت هيبقى جواها 4 فانكشن رئيسين

OnGet , OnPost , OnDelete , OnPut

طيب المفروض بقه عشان اشغل اى واحده فيهم اعمل ايه ؟؟

اكون باعت فى الهيدر بتاع الريكيوست انى عاوز اكلم ال Get او عاوز اكلم ال POST او عاوز اكلم ال Delete او عاوز اكلم ال PUT

طب الموضوع ده يحصل اذاى ؟؟

تعالى اوريك فى الكود احسن و بعد كده هشرحلك

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

asp-page-handler

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

ده اتربيزت بحطه الزرارير هو الى بيعمل مابينج مع اسامى الفانكشن و يشغلها على اساس الاسم الى انا باعته معاه و عشان تفهم بص يا سيدى اول حاجه احنا المفروض عشان نعمل Get , Post , Delete , Put اننا نعمل Form مهمه الفورم دى انى اقول للكود فيها لما ادوس على زرار الاضافه فهو ياخد كل البيانات الى فى الفورم و يروح يبعتها للفانكشن الخاصه بالاضافه و لما اقول اعمل تعديل يروح يعمل تعديل و لما اقوله روح امسح هيروح يمسح

طيب هو هيفرق اذاى و هو رايح من الفورم فى الفرونت ايند للباك اند ؟؟

عن طريقه attribute انا بحطه فى الفورم خاص بال Razor بيعرف الكود انه رايح للفانكشن الفلانيه

يعنى على سبيل المثال

انا هعمل Get فبقوله فى الفورم Get طيب لو رايح اعمل Post بقوله Method = Post

طيب لو حبيت اعمل Delete او Update هنا بقه الدنيا هتقف لان مفيش فى ال HTML الا Get , Post

فراحوا عاملين فى ال Razor اتربيوت خاص بيها بيفرق بين كله و بعضه

فمثلا انا عاوز اعمل POST , Delete , Update , Get

بروح محدد اسم معين لكل واحده فيهم و احطه جوه اتربيوت داخل الزراير بتاعتهم جوه ال Form و بخليهم كلهم Mehod=Post عشان موجعش دماغى و الاختلافى ال Attribute الى اسمه asp-page-handler هو بقه بياخد الاسم ده و يدور عليه فى ال CS و اول ما بيشوفه بيروح رايح مشغل الفانكشن دى

ده مثال بقه عشان تفهم ايه الى بيحصل

ال index.cshtml.cs

namespace WebAppl.Pages

{

public class IndexModel : PageModel

{

private readonly ILogger<IndexModel> \_logger;

public IndexModel(ILogger<IndexModel> logger)

{

\_logger = logger;

}

public string Message { get; set; } = "Initial Request";

public void **OnGet**()

{

Message = "Form Get";

}

public void **OnPost**()

{

Message = "Form Posted";

}

public void **OnPostDelete**()

{

Message = "Delete handler fired";

}

public void **OnPostEdit**()

{

Message = "Edit handler fired";

}

public void **OnPostView**()

{

Message = "View handler fired";

}

}

}

ذى ما نت شايف فوق انا سميت كل الفانكشنز On + Mehod type from form + HandlerName

الهاندلار نيم ده هو الاسم الى انا ببعته فى الاتربيوت فى ال HTML من ال Form فى ال asp-page-handler

يعنى لو الفورم ال Mehod = POST و ال Asp-page-handler = Badawy

يبقى فالكود السى شارب عشان الاتش تى ام ال يعمل ماب معاها و يششغلها لازم ابقى مسميها

**OnPostBadawy**

ولو الفورم ال Mehod = GET و ال Asp-page-handler = Ma7shy

**OnGetMa7shy**

ملحوظه مهمه بقه

ال OnGet دى بترن بمجرد ما الصفحه يتعلماه رن دى فانكشن شغاله اوتوماتيك لو عاوز تشيلها عشان متشتغلش مفيش مانع بس عادى يعنى دى تحط فيها ال Intial Values

و ادى كود ال HTML عشان تفهم كلامى

@page

@model IndexModel

@{

ViewData["Title"] = "Home page";

}

<div class="row">

<div class="col-lg-1">

<form **asp-page-handler**="**edit**" method="**post**">

<button class="btn btn-primary">Edit</button>

</form>

</div>

<div class="col-lg-1">

<form **asp-page-handler**="**delete**" method="**post**">

<button class="btn btn-primary">Delete</button>

</form>

</div>

<div class="col-lg-1">

<form **asp-page-handler**="**view**" method="**post**">

<button class="btn btn-primary">View</button>

</form>

</div>

<div class="col-lg-1">

<form **asp-page-handler**="" method="**post**">

<button class="btn btn-primary">post</button>

</form>

</div>

</div>

<h3 class="clearfix">@Model.Message</h3>

هتلاحظ انك لما بتدوس على اى زرار ال View مثلا فهو بيروح مغير ال Route من

<https://localhost:44370/>

الى

<https://localhost:44370/?handler=view>

و ده معناه ان الى انا asp-page-handler ده بيغير فى ال Route و يزود عليه كويرى سترينج اسمها

Handler و بتاخد القيمه الى جوه ال asp-page-handler و بتتكتب فى اللينك على انها كويرى سترينج

طيب لو انا بقه عاوز اخليها بدل ما تتكتب كويرى تسرينج اخليها تتكتب سلاش view ذى كده

<https://localhost:44370/view>

اعمل ايه ؟؟

كده اروح اغير طريقه الكتابه فى ال Page من الطريقه الافتراضيه للطريقه الى انا عاوزها و ده عن طريق انى احدد ال Route بأيدى

طب و الكلام ده يتعمل اذاى ؟؟

الكلام ده اعمله لو بانى اكتبله الكلام ده فى ال HTML جنب ال Page@

هنا انا عاوز ابعتله قيمه ال Hanlder بس و ممكن ابعتها او مبعتهاش يعنى اختياريه يبقى اعملها كده

**@page "{handler?}"**

@model IndexModel

@{

ViewData["Title"] = "Home page";

}

<div class="row">

<div class="col-lg-1">

<form **asp-page-handler**="edit" method="post">

<button class="btn btn-primary">Edit</button>

</form>

</div>

<div class="col-lg-1">

<form **asp-page-handler**="delete" method="post">

<button class="btn btn-primary">Delete</button>

</form>

</div>

<div class="col-lg-1">

<form **asp-page-handler**="view" method="post">

<button class="btn btn-primary">View</button>

</form>

</div>

<div class="col-lg-1">

<form **asp-page-handler**="" method="post">

<button class="btn btn-primary">post</button>

</form>

</div>

</div>

<h3 class="clearfix">@Model.Message</h3>

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

**الهرى الى فوق ده كله ممكن احطه جوه فورم واحده و اريح دماغى**

@page "{handler?}"

@model IndexModel

@{

ViewData["Title"] = "Home page";

}

<div class="row">

<div class="col-lg-1">

<form method="post">

<button **asp-page-handler**="edit" class="btn btn-primary">Edit</button>

<button **asp-page-handler**="delete" class="btn btn-primary">Delete</button>

<button **asp-page-handler**="view" class="btn btn-primary">View</button>

<button **asp-page-handler**="" class="btn btn-primary">post</button>

</form>

</div>

</div>

<h3 class="clearfix">@Model.Message</h3>

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

Send Values With Request

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

بفرض انى عاوز ابعت ID هعمل ايه ؟؟

هعمل input و اسمه ID و و اروح فى الاضافه و التعديل والحزف اعرفهم ان الفانكشنز بتعتهم هتاخد براميتار نوعه Int و اسمه ID فكده خلاص هيتبعتله و بمجرد ما دخلت اى رقم فى ال input و دوست على اى زرار هيروح واخد الرقم و بعته فى الكويرى سترينج

عشان نوضح هيبقى الكود كده فى ال HTML

@page "{handler?}"

@model IndexModel

@{

ViewData["Title"] = "Home page";

}

<div class="row">

<div class="col-lg-1">

<form method="post">

<button **asp-page-handler**="edit" class="btn btn-primary">Edit</button>

<button **asp-page-handler**="delete" class="btn btn-primary">Delete</button>

<button **asp-page-handler**="view" class="btn btn-primary">View</button>

<button **asp-page-handler**="" class="btn btn-primary">post</button>

<input type="text" **name="id"** value="" />

</form>

</div>

</div>

<h3 class="clearfix">@Model.Message</h3>

**و كده فى السى شارب و ليكن هغير ال Edit**

public void OnPostEdit(int id)

{

Message = "Edit handler fired ID = " + id;

}

فهنا ضيفت براميتار و بينته فى الماسيدج عشان اشوفه اتبعت ولا لأ

لما جيت اجرب بقه لقيته فعلا بعته فى الماسيدج و متكتبش فى ال Route على الرقم انى مقولتوش فى ال Route انى هبعته و ده لانى باعته من ال Input فمتكتبش فى ال Route

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

انما لو بعته بالشكل ده بقه جوه asp-route-id او asp-route-fegl هتلاقيه هيتكتب فى كويرى سترينج على طول

@page "{handler?}"

@model IndexModel

@{

ViewData["Title"] = "Home page";

}

<div class="row">

<div class="col-lg-1">

<form **asp-page-handler="delete" asp-route-id="10" asp-route-Fegl="Garger"** method="post">

<button class="btn btn-default">Delete</button>

</form>

</div>

</div>

<h3 class="clearfix">@Model.Message</h3>

ال Cs بقه فى فانكشن ال delete المفروض هبعت براميتار اسمه id و اخر اسمه Fegl عشان يعرف يماب و ياخد قيمه ليه يستعملها جوه الفانكشن فهيبقى كده

public void OnPostDelete(**int id , string Fegl**)

{

Message = "Delete handler fired ID = " + id + " Nos Kelo : " + Fegl;

}

اللينك الى هيبقى ظاهر بعد ما ادوس على زرار المسح هيبقى كده

<https://localhost:44370/delete?id=10&Fegl=Garger>

طيب لو انا مش عاوز ال ID , Fegl دول يظهروا فى ال ال Route المفروض اعمل ايه ؟؟

المفروض اروح فى الروت عند ال @Page اعرفه ان هيجيله ID,Fegl و احدد لكل واحد نوعه و هل اختيارى ولا اجبارى

الاختيارى بحط بعده علامه استفهام و الاجبارى مبحطش علامه استفهام

عمتا علامه استفهام (؟) فى السى شارب معناها الحاجه دى اختياريه او تنفع تاخد Null

فهعمل الروتينج كده بقه عشان مفيش حاجه تظهر ككويرى سترينج يا كبير و اجرب الدنيا

@page "{handler?}/{id?}/{Fegl?}"

هتلاقى الدنيا شغاله ذى الفل و لكن اللينك او الروت بقى شكله فوق كده

<https://localhost:44370/delete/10/Garger>

و كده اتبعت على انه روت اهه

شوف بقه اللينك ده و هتفهم منه كويس جدا الكلام الى فوق ده

<https://www.learnrazorpages.com/razor-pages/handler-methods>

طيب انا ممكن صحيح ابعت ال ID و Fegl من ال input و مش هحتاج اكتبهم فى ال Route لانهم هيعملوا راوتينج بشكل اوتوماتيك على فكره D:

@page "{handler?}"

@model IndexModel

@{

ViewData["Title"] = "Home page";

}

<div class="row">

<div class="col-lg-1">

<form **asp-page-handler**="delete" method="post">

<button class="btn btn-primary">Delete</button>

<input type="type" name="id" value="" />

<input type="type" name="Fegl" value="" />

</form>

</div>

</div>

<h3 class="clearfix">@Model.Message</h3>

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

namespace WebAppl.Pages

{

public class IndexModel : PageModel

{

public string Message { get; set; } = "Initial Request";

public void OnPostDelete(int id , string Fegl)

{

Message = "Delete handler fired ID = " + id + " Nos Kelo : " + Fegl;

}

}

}

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

**لحد هنا احنا كويسين اوى بس لو انا بقه هبعت اوبجيكت من الفورم للسى شارب دى تتعمل اذاى بقه ؟؟**

سهله جدا

كل الى هعمله كلاس جديد جواه بروبارتى ID, Fegl و هروح عامل ان البراميتار الى الفانكشن بتاخده حاجه من نوع الكلاس الجديد ده و بما انى هبعت الحاجات دى من ال input الى فى ال form هيحصل مابينج بين الاوبيجت الى الفورم بعتاه لان الفورم بتبعت القيم على صوره Object فلو الاوبجيكت الى مبعوت من ال form فيه نفس البروبارتيس الى فى الكلاس الجديد فهيحصل مابينج و الداتا هتتبعت عادى

و ادى شكل الاتش تى ام ال

@page "{handler?}"

@model IndexModel

@{

ViewData["Title"] = "Home page";

}

<div class="row">

<div class="col-lg-1">

<form **asp-page-handler**="delete" method="post">

<button class="btn btn-primary">Delete</button>

<input type="type" name="**id**" value="" />

<input type="type" name="**Fegl**" value="" />

</form>

</div>

</div>

<h3 class="clearfix">@Model.Message</h3>

**و الى سى اس اس**

namespace WebAppl.Pages

{

**public class MyClass**

**{**

**public int id { get; set; }**

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

**}**

public class IndexModel : PageModel

{

private readonly ILogger<IndexModel> \_logger;

public IndexModel(ILogger<IndexModel> logger)

{

\_logger = logger;

}

public string Message { get; set; } = "Initial Request";

public void OnPostDelete(**MyClass Data**)

{

Message = "Delete handler fired ID = " + Data.id + " Nos Kelo : " + Data.Fegl;

}

}

}

MB

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

View Data

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

ال ViewData هو عباره عن كونتينار اقدر اشيل فيه اى داتا من اى نوع عادى خالص بيوصل الداتا من ال PageModel لل View و العكس يا كبير يعنى بيعمل داتا بيندينج

ال ViewData ده هو Dictionary of Object يعنى يقدر يشيل كذا أوبجيكت مع بعض عادى

طيب تعالى اوريك بقه ال ViewData هياخد من ال PageModel اذاى فى الكود الى جاى ده

دى صفحه ال PageModel

namespace WebAppl.Pages

{

public class Course

{

public int id { get; set; }

public string Title { get; set; }

public string Price { get; set; }

}

public class IndexModel : PageModel

{

private readonly ILogger<IndexModel> \_logger;

public IndexModel(ILogger<IndexModel> logger)

{

\_logger = logger;

}

public void OnGet()

{

//Simple Data

ViewData["Name"] = "Mahmoud";

ViewData["Age"] = 25;

//Complex Data (Object)

Course MyCourse = new Course

{

id = 1,

Title = "ASP.Net Core",

Price = "500 $"

};

ViewData["CourseObj"] = MyCourse;

}

}

}

هتلاحظ انى اول حاجه حطيت فى الفيوداتا الخاصه بالاسم اسمى و سنى و ده داتا بسيطه مش محتاجه حاجه و لكن لما جيت بقه هحط حاجه كومبليكس الى هو داتا بتاعت كلاس ال course فكونت اوبجكت و بعته فيه و ده برضوا مش حاجه صعبه الدنيا ايزى يعنى

و لكن الاختلاف فى الاستقبال بقه فى ال HTML بص يا سيدى انت فى الحاجات البسيطه هتقوله فقط @ViewData[“index”] يعنى من الاخر نفس الشكل الى فى ال pageModel و فقط بحط قبله @

طيب لو هتعامل مع حاجه كومبليكس بقه من نوع Course ده بقه يا سيدى بستقبل الداتا الى راجه دى فى اوبجيكت و بعد كده بطلع الداتا من الاوبجيكت عادى و طبعا عشان اعرف احط الدتا فى اوبجيكت من نوع كلاس Course بتضطر اعمل Casting لل ViewData الى نفس نوع الاوبجيكت الى هحطه فيه و الى هو Course و ادى الكود

@page "{handler?}"

@model IndexModel

@{

ViewData["Title"] = "Home page";

}

<div>

<p>My Name is : @ViewData["Name"] </p>

<p>My Age is : @ViewData["Age"] </p>

</div>

@{

**Course Data = (Course) ViewData["CourseObj"];** // Object Casting to Course type

**Course Data\_2 = ViewData["CourseObj"] as Course;**

}

<div>

<p>Course ID : **@Data.id** </p>

<p>Course Title : **@Data.Title** </p>

<p>Course Price : **@Data.Price**</p>

</div>

MB

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

View Data Attribute

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

ده يا سيدى طريقه جديده لكتابه ال viewData و دى عامله ذى اى اتربيوت فى ال C# الى هو لما بكتبه فوق اى حاجه و ليكن هكتبه فى property فالبروبارتى ده بقه كده ViewData عادى خالص

و بالتالى شكل الكتابه الجديد هيبقى كده انا هكتب نفس الكود القديم بس بطريقه ال Attribute

namespace WebAppl.Pages

{

public class Course

{

public int id { get; set; }

public string Title { get; set; }

public string Price { get; set; }

}

public class IndexModel : PageModel

{

**[ViewData]**

public string Name { get; set; }

**[ViewData]**

public int Age { get; set; }

**[ViewData]**

public Course CourseObj { get; set; }

public void OnGet()

{

//Simple Data

Name = "Mahmoud";

Age = 25;

//Complex Data (Object)

Course MyCourse = new Course

{

id = 1,

Title = "ASP.Net Core",

Price = "500 $"

};

CourseObj = MyCourse;

}

}

}

ده بقه لينك هيفهمك جدا جدا

<https://dotnettutorials.net/lesson/viewdata-asp-net-core-mvc/>

MB

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

Action Result

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

**ActionResult** دى يا سيدى انا بقدر ارجع فيها حاجات كتير منها **Json , View , HttpRequest و** ممكن اعمل فيها **RedirectToPage** عادى و كمان ممكن ارجع **فيها Particial Page**

فمثلا انا هنا هرجع فيها صفحه الايرور و نشوف هتشتغل ولا لأ :D

namespace WebAppl.Pages

{

public class IndexModel : PageModel

{

public ActionResult OnGet()

{

return RedirectToPage("Error");

}

}

}

ممكن ارجع فيها أوبجيكت اهه برضوا

namespace WebAppl.Pages

{

public class Course

{

public int id { get; set; }

public string Title { get; set; }

public string Price { get; set; }

}

public class IndexModel : PageModel

{

public Course CourseObj { get; set; }

public ActionResult<Course> OnGet()

{

Course CourseObj = new Course {id = 1,Title = "ASP.Net Core",Price = "500 $"};

return CourseObj;

}

}

}

عمتا هناخدها بشكل تفصيلى مع ال API

MB

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

ActionResult VS IActionResult

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

ActionResult : دى الكلاس الاصليه لما بحب ارجع حاجه برجعها فيها

IActionReult : ده الانترفيس يا كبير ده بقه معمول عشان لو انا عاوز اعمل Cusomization و اعملى كلاس خاصه بيا فيها فانكشنالتى زياده او شويه تعديلات فبورث من الانترفيس و اعد اكتب بقه و اعدل فى الفانكشنالتى بالشكل الى يناسبنى

MB

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

ViewComponent

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

هو كومبونانت بقدر استخدمه فى كذا مكان و بقدر اكتب فيه فانكشنز معقده كتير و استخدمها فى اماكن كتير نفس فكره الكومبونانت الى فى انجيولار انى اعمل كومبونانت و اناديه فى اى مكان براحتى و بيبقى معاه الفانكشنز بتعته

من الاخر كده لما تلاقى ان الدنيا هتبقى تقيله و هتستعمل ال Syncronization و هتكتب كود فشيخ يبقى استعمل ال ViewComponent و فكك من ال Partial Pages العاديه

تعالى بقه نديك مثال

بص انا بالصلاه على النبى كده هروح اجيب داتا من ال JsonPlaceHolder و هقرأها و احطها فى الكومبونانت يلا بينا

أول حاجه انا عاوز اجيب الداتا من اللينك ده

<https://jsonplaceholder.typicode.com/users>

فأول حاجه هبص على الاوبجيكت بتاعه و على اساسه هكون الكلاسات الخاصه بال ViewModel بقه عشان اعرف اتعامل مع الداتا دى بسهوله بعد كده

فأول حاجه هروح اعمل الكلاسيز الى هشيل فيها الداتا بعد كده كل كل واحده فى فايل ما و بالتالى انا المفروض هعمل فولدر اسميه ViewModel مثلا

اختصارا للوقت و المجهود انا هعمل كل الكلاسات الى هحتاجها فى استقبال الداتا فى فايل واحد هسميه DomianVM و هتبقى كده

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace WebAppl.ViewModels

{

public class users

{

public int id { get; set; }

public string name { get; set; }

public string username { get; set; }

public string email { get; set; }

public string phone { get; set; }

public string website { get; set; }

public address address { get; set; }

public company company { get; set; }

}

public class address

{

public string street { get; set; }

public string suite { get; set; }

public string city { get; set; }

public string zipcode { get; set; }

public geo geo { get; set; }

}

public class geo

{

public string lat { get; set; }

public string lng { get; set; }

}

public class company

{

public string name { get; set; }

public string catchPhrase { get; set; }

public string bs { get; set; }

}

}

بعد كده هروح اعمل فولدر و اسميه Services و ده هيبقى فيه ايه بقه ؟؟

ده هيبقى فيه ال Services الى بعملها للمشروع كله و دى بتبقى متشافه جلوبال فاكر السيرفيسيس الى فى الانجيولار اهه دى زيها

المهم هنا بقه بعمل فايلين يعنى 2 فايل للسيرفيس بحطهم جوه فولدر اسمه **services/Users**

واحد اسمه IUsersServices و التانى UsersServices الى هو من الاخر فايل بعمله انترفيس و فايل تانى بعمله للامبلمنتيشن

أول حاجه فايل الانترفيس هكتب فيه الفانكشنز الى انا محتاجها و شكرا على كده

namespace WebAppl.Services.Users

{

public interface IUsersService

{

public Task<List<users>> GetUsersDataAsync();

public Task<users> GetUserDataAsync(int ID);

}

}

اما السيرفيس ذات نفسها بقه هتبقى كده

namespace WebAppl.Services.Users

{

public class UsersService : IUsersService

{

public async Task<users> GetUserDataAsync(int ID)

{

using (var client = new HttpClient())

{

var EndPoint = $"https://jsonplaceholder.typicode.com/users/{ID}";

var JsonData = await client.GetStringAsync(EndPoint);

return JsonConvert.DeserializeObject<users>(JsonData);

}

}

public async Task<List<users>> GetUsersDataAsync()

{

using (var client = new HttpClient())

{

var EndPoint = @"https://jsonplaceholder.typicode.com/users";

var JsonData = await client.GetStringAsync(EndPoint);

return JsonConvert.DeserializeObject<List<users>>(JsonData);

}

}

}

}

ذى ما انت شايف بعمل الفانكشن و اقوله انها async و جواها بقوله استخدم الكلاس الى اسمه HttpClient عشان تتعامل مع ال HttpRequest , HttpResponse و يبقى التعامل مع ال API الى معاك من اللينك الى انت جبته ده سهل و اروح رايح مشغل بقه فانكشن ال GetStringAsync عشان تجيبلى الداتا و فى نفس الوقت اقول للكود await الى هو يعنى كله بستنى لحد ما السر ده يتنيل على عينه يخلص و بعد كده رجع الداتا الى انت جبتها بقه بس طبعا بعد ما تكون انت حولتها من سترينج لاوبجيكت لان الداتا قبل ما نعملها Deserialize بتبقى سترينج ولازم تحولها بال JsonConvert لاوبجيكت يا كبير

ملحوظه مهمه Using () {} دى بنعملها عشان تعمل ديستروى للاوبجيكت لما تخلص عشان تمسح الجاربيدج كوليكشن من الاخر كانى بعمل unsubscribe بتاعت انجيولار :D

كده خلصنا السيرفيس يبقى نخش على الخطوه الى بعديها بقه و هى انى هروح احط السيرفيس دى جوه ال startup فى **ConfigureServices** ذى ما بعمل مع السيرفيس فى ال AppModule فى الانجيولار برضوا هههههههههههه و هتبقى بالشكل ده

public void ConfigureServices(IServiceCollection services)

{

services.AddRazorPages();

// Register Service to providers in Config

**services.AddTransient<IUsersService,UsersService>();**

}

كده انا سجلت السيرفيس الى انا عاملها اقدر استخدمها بقه فى اى كومبونانت فهروح اعمل دلوقتى بقه حاجتين

الحاجه الاولى انى هعمل Component و الحاجه التانيه انى هعمل ViewComponents

دول بقه ذى بالظبط ال Component. html و ال Component.ts فى الانجيولار

ال ViewComponent ده هو الى ذى ال ts فايل بالظبط فده بكتب فيه البيزنيس بتاع الكومبونانت يعنى ده لكتابه السى شارب و اتسمى بال view لان بيعمل ريتيرن ل View الى هو بيانات صفحه ال Component

**ملحوظه هامه جدا**

طيب دلوقتى بقه الفريم ورك بيدور على الكومبونانت فى اماكن معينه ألا و هى يا باشا فولدر ال Pages

فانا دلوقتى جوه فولدر ال **Pages** هعمل فولدر اسمه **Components** و ده بقه الى هيبقى شايل كل **الكومبونانتس** و كل ال **ViewComponents** بتعتها و بما انى هعمل كومبونانت لليوزر فهروح عامل اول حاجه فولدر اسمه **Users** كده الباث اصبح Pages/Components/Users

جوه الفولدر بقه

الفايل الاول اسمه UsersViewComponent ده الى هيبقى فيه بيزنس الكومبوناتس بتاعت ال Users

الفايل التانى UsersComponent ده الى هيبقى فيه ال HTML الخاص بلسته المستخدمين

و ادى ال UsersViewComponent هعمله كلاس عادى خالص بس بيورث من ال ViewComponent و ده شغله فقط على ارجاع اللسته و مينفعش تعمل فيه تاسك تانيه فيها InvokeAsync لان دى لازم تبقى فانكشن واحده وبس اسمها InvokeAsync و لو فانكشنز تانيه تبقى فانكشنز بتخدمها من الاخر

namespace WebAppl.Components.Users

{

public class **Users**ViewComponent: **ViewComponent**

{

private IUsersService \_UsersService;

public UsersViewComponent(IUsersService UsersService)

{

\_UsersService = UsersService;

}

public async Task<IViewComponentResult> InvokeAsync( )

{

List<users> Users = await \_UsersService.GetUsersDataAsync();

return View(Users);

}

}

}

ذى ما انت شايف يا سيدى دى كده الفانكشن الى انا برجع بيها الداتا الى هبعتها لل View و اول حاجه بعملها فيها انى بعمل للسيرفيس Dependency Injection عشان اقدر استعمل الفانكشنز بتعتها فى الفايل ده و ده عن طريق انى بعمل فاريبول من نوع الانترفيس او الكلاس بتاعه و بعد كده اعمل ارجيومينت للكونستراكتور بنوع السيرفيس و ده هيبقى شايل الداتا بتاعت الكلاس او الانترفيس اصلا و بعد كده بحط الى جاى من الارجيومينت جوه الفاريبول البرايفيت الى انا عامله و ابدأ بقه استخدم الحاجه البرايفت بتاعتى يا كبير

تانى فانكشن انا بعمل حاجه بترجع **تاسك من نوع IViewComponentResult** عشان تبقى بترجع View يترزع فى الكومبونانت بتعها بقه ميقولش لأ و بقوله ان اسم الفانكشن **InvokeAsync** و دى فانكشن أساسيه لازم اكتبها لو متكتبتش بنفس الشكل يطلعلك ايرور على طول جواه بقه بجيب البيانات من السيرفيس و احطها فى لسته و بعد كده ببعت اللسته دى فى View عشان اقدر اعرضها فى الكومبونانت بقه بعد كده

تانى خطوه بقه اروح اعمل فايل ال Componet الخاص باظهار بيانات لسته المتسخدمين فعمله داخل **فولدر** اسمه **Users** الى جوه فولدر ال **Components** و بعد كده جوه **الفولدر** **user** ده هعمل فايل اسمه **Default.cshtml** و هيبقى نوع الفايل ده من نوع Razor View جواه بقه هكتب كود يخلينى اعرف اوصل للكومبونانت و اجيب منه البيانات الى جوه ال View الى الفانكشن مرجعاها و **Default** دى لازم يبقى اسم اى فايل بتاع الكومبونانت عشان يشتغل و دى حاجه بتفرضها علينا الفريم ورك برضوا و هو بيماب مع الفيوكومبونانت بتاعه عادى لا تقلق لان هو بيعرف من حاجه معينه و هو ان

ال ViewComponent بيبقى اسمه **Users**ViewComponent و اسم الفولدر الى فيه ال component او ال Razor View اسمه **Users** فبيروح عامل مابينج بينهم و يعرف ان الاتنين دول بتوع بعض فيشتغلوا مع بعض

@\*@model namespace.your class name\*@

@\*name space\*@

@model List<WebAppl.ViewModels.users>

<h1> List Of Users</h1>

<ul>

@foreach (var Item in Model)

{

<li> @Item.username </li>

}

</ul>

ذى ما انت شايف اول حاجه بقوله بص فى ال model دى انا عاوز استقيل حاجه نوعها List من ال users و بعد كده طالما لسته بروح عامل Foreach عشان اجيب البيانات فى الموديل احطها فى ال Item و بعد كده اعرض الى انا عاوز فى الكومبونانت بقه

لحد كده انا كده كونت الكومبونانت ناقص بقه انى اظهره فى صفحه عندى بقه اعملها اذاى دى ؟؟

دى كانى هاخد السليكتور بتاعه احطه فى صفحه تانيه و العمليه دى بتتم بحاجه بسيطه يعنى انا مثلا دلوقتى عاوز اعرض الكومبونانت ده فى ال Home page index فهروح رايح جواها و كاتب كده

@page

@model IndexModel

@{

ViewData["Title"] = "Home page";

}

**@await Component.InvokeAsync("Users")**

خد بالك برضوا انى عملت هنا await و قولتله روحلى على الكومبونانت و شغلى الفانكشن الى اسمها InvokeAsync على الكومبونانت الى اسمه **Users** و شكرا على كده

طيب لو عاوز اجيب بقه تفصيله معينه عن اى User كده هروح اعمل كومبونانت تانى و يبقى الكومبونانت ده اسمه User و جواه فايلين واحد للفيوكومبونانت وواحد للكومبونانت نفسه

ال UserViewComponent هيبقى كده

namespace WebAppl.Pages.Components.User

{

public class UserViewComponent : ViewComponent

{

private IUsersService \_UserService;

public UserViewComponent(IUsersService UserService)

{

\_UserService = UserService;

}

public async Task<IViewComponentResult> InvokeAsync(int ID)

{

users User = await \_UserService.GetUserDataAsync(ID);

return View(User);

}

}

}

ذى ما انت شايف فى باراميتار فى الانفوك من نوع انتيجر و مسميه ID ده عشان عن طريق اى دى المستخدم اجيب بياناته

عندك بقه تانى حاجه انى اعمل الكومبونانت نفسه فهيبقى كده default.htmlcs

@model WebAppl.ViewModels.users

<table class="table">

<thead>

<tr>

<th scope="col"> username </th>

<th scope="col"> address </th>

<th scope="col"> company </th>

<th scope="col"> email </th>

<th scope="col"> phone </th>

</tr>

</thead>

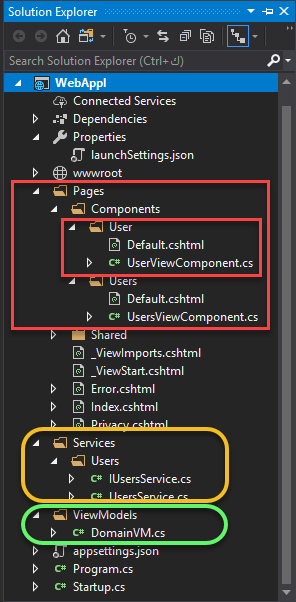
<tbody>

<tr>

<td scope="col"> @Model.username </td>

<td scope="col"> @Model.address.city </td>

<td scope="col"> @Model.company.name </td>

 <td scope="col"> @Model.email </td>

<td scope="col"> @Model.phone </td>

</tr>

</tbody>

</table>

كده بقه مش باقى غير انى اروح اعرضه فى Home Page Index و عشان اعرضه هناك هكتب السطر ده ياكبير

@await Component.InvokeAsync("User",1)

لاحظ انى بعتله رقم 1 و ده بديل عن ال ID و بكده هيجيب تفاصيل بيانات المستخدم الى رقم الاى دى بتاعه 1 و عبد كده هنبقى ناخد الارقام دى بشكل Dynamic من بيدجات تانيه و كده ففاكس دلوقتى ليه ده تجربه و خلاص يعنى

ده شكل الفوردرات بقه الى انا عملتها للحاجات دى

MB

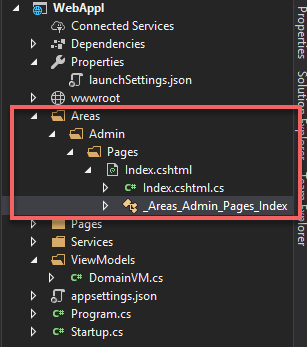
################################################################

Area and it’s Routing and LayOut for Area

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

ال Areas دى يا صديقى المكان الى بتعمل فيه صفحاتك براحتك يعنى تعمل فيه قسم الادمن قسم الكورسات قسم مش عارف ايه بس عشان تبقى فاهم برضوا الدنيا فى ال Razor ماشيه بهيراركى و استراكتشار معين فانت اول حاجه عشان تعمل بيدجات جديده هتروح رايح عامل فولدر اسمه Areas طيب انا دلوقتى عاوز ابتدى اعمل صفحات الادمن فهروع عامل فولدر اسمه Admin طيب بما انى لازم اعلم للادمن صفحه خاصه بيه فهروح عامله فولدر اسمه Pages جوه الفولدر ده بقه هحط ال Razor Pages

الباث هيبقى عشان اوصل لصفحه الادمن كده ~/Areas/Admin/Pages/index.htmlcs



هتخش بقه على ال index.cshtml هتلاقى ان الموديل بتاعه بيستخدم الليبرارى بتاعت الاندكس الى جوه الاريا الى هو من الاخر مكتوبه كده

@model WebAppl.Areas.Admin.Pages.IndexModel

انا بصراحه شايف انها كده حلوه و كويسه لكن فى ناس تقولك انا كل مره هروح اكتب النيم سبيس هنا بالشكل ده طب منا احط النيم سبيس فى ViewImport احسن

فهروح عامل الفيو امبورت و احط فيه النيم سبيس ليصبح شكل ال \_ViewImport شكله كده

@namespace WebAppl.Areas.Admin.Pages

و كده يا معلم اصبح ليك الحق انك جوه ال index.cshtml تنادى على ال indexModel بالشكل ده

@model IndexModel

**اذا ايه اهميه ال \_ViewImport ؟؟**

بص اهميته انى مثلا ممكن اكون عاوز اعمل 100 import فهل ده طبيعى انى اعمله فى ملف ال HTML ؟؟

لأ طبعا و عشان كده ظهر الفايل الى اسمه \_ViewImport ده عشان الم فيه كل ال imports الخاصه بال indexModel و يبقى التركيز فى انى اكتب الاتش تى ام ال هنا و خلاص و اظهر الدتا و شكرا

طيب انا عاوز اجرب ال indexModel اشوفه شغال و لا ايه فهروح اكتب جواه اى نيله كده

@page

@model IndexModel

<h1> Hello From Admin Page </h1>

طيب كده بقه مكتوب فيها حاجه اهه اعرفها بيها بس ناقص اعرف اذاى اوصل للمكان بتعها اصلا

طب انا هوصل ليها اذاى ؟؟ عن طريق لينك يا اسطى يبقى هعملها راوتينج

اذا انا هروح للصفحه الرئيسيه الى فى ال Home Page و احط لينك فيه Route يودينى لصفحه الادمن و ادى Home Page index يا معلم

@page

@model IndexModel

@{

ViewData["Title"] = "Home page";

}

<button class="btn btn-dark">

<a **asp-page**="/Index" **asp-area**="Admin"> Admin </a>

</button>

مجرد بتقوله ان الصفحه اسمها index و فى ال area الى اسمها admin

طيب لو عاوز تعمل صفحه تانيه باسم مثلا addUsers هل ينفع ؟؟

ايوه اكيد

انا مثلا عملتها و كتب جواها كده

@page

@model AddUsersModel

<h1> Hello From AddUsers Page </h1>

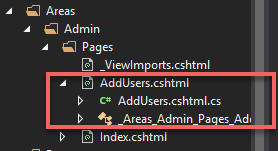
و بتعملها راوت عادى خالص ذى الى فوق بس بتغير اسم الصفحه فهيبقى الزرار مكتوب كده فروحت ضيفتلها الزرار الى بيودى ليها و كتبته كده

<button class="btn btn-dark">

<a **asp-page="/AddUsers" asp-area="Admin"**> AddUsers </a>

</button>

و شكل الفايلات و الفولدرات كده

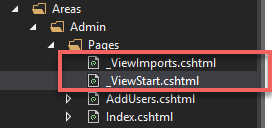


طيب انا لقيت انى كل اما بدخل على صفحه الناف بار الى فوق بيختفى و معنى كده ان ال layout كده انا مش حاطه فحل المشكله دى ايه ؟؟

اعمل حاجه اسمها ViewStrat دى مهمتها انها بتعرض فايلات ال HTML المشتركه ذى فايل ال \_LayOut

فانا عشان اشغل الناف بار الاساسى فى المشروع فى المكان ده هروح رايح عامل ال ViewStart فى فولدر ال Pages فى ال Admin

يصبح شكل الفايلات بقه كده



جوه ال \_viewStart بحدد بقه انهى \_layOut يتم استخدامه لان انا ممكن اروح اعمل 100 layout و استخدم الى على مزاجى و ده فعلا الى بيحصل لما يكون هخش على الادمن موديول بس انا هنا عاوزه يستخدم ال LayOut بتاع عامه الشعب فهكتب فى جوه الفايل بتاع ال \_ViewStart.cshtml كده

@{

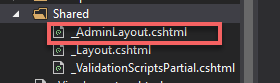
Layout = "\_Layout";

}

طيب لو انت عاوز تعمل LayOut جديد تعمل ايه ؟؟

بتروح على فولدر ال Shared و تاخد كوبى من القديم و تعمل واحد جديد باسم جديد و ليكن هنسميه \_AdminLayout

\_



و هتعدل جواه براحتك بقه و تيجى فى ال ViewStart بتاع ال admin تكتب اسمه ال layout كده بقه

@{

Layout = "\_AdminLayout";

}

**يبقى ايه لازمه ال \_ViewStart ؟؟**

دى من الاخر بتتحط جوه Module معين و بتعرفه بال layout الخاصه بيه و بيتحط جواها اسم ال Layout بتاعت الموديول

و ال Layout دى بتبقى ذى ال Navbars و ال Footer يعنى الحاجات الى ثابته فى اغلب الموقع او اوقات بيسموها ال Master pages

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

{

"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 Home Page indexModel فتعالى اوريك هيتقرأ اذاى

namespace WebAppl.Pages

{

public class IndexModel : PageModel

{

private IConfiguration Configuration { get; }

public string ConnString { set;get;}

public IndexModel(IConfiguration configuration)

{

Configuration = configuration;

}

public void OnGet()

{

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

ConnString = Configuration["ConnectionString"];

}

}

}

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

و عشان اتاكد ان البيانات جت فعلا فانا فعلا حبيت اظهر الكلام ده فى ال Html فعملت الكود ده فى ال index.cshrml

@page

@model IndexModel

@{

ViewData["Title"] = "Home page";

}

<h1>

@ViewData["ProjectName"]

</h1>

<hr />

<h1>

@Model.ConnString

</h1>

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

ViewData["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 WebAppl

{

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 AppSettingCls.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

فى ال index.cshtml.cs

namespace WebAppl.Pages

{

public class IndexModel : PageModel

{

private IConfiguration Configuration { get; }

public string DefaultConnString { set;get;}

public string RealName { set; get; }

public string DefaultName { set; get; }

public IndexModel(IConfiguration configuration)

{

Configuration = configuration;

}

public void OnGet()

{

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

DefaultConnString = ConString.DefaultConnStr;

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

RealName = ProjectName.RealName;

DefaultName = ProjectName.DefaultName;

}

}

}

و ال Html بتعها هيبقى كده بقه

@page

@model IndexModel

@{

ViewData["Title"] = "Home page";

}

<h1>

@Model.RealName

</h1>

<hr />

<h1>

@Model.DefaultName

</h1>

<hr />

<h1>

@Model.DefaultConnString

</h1>

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

MB

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

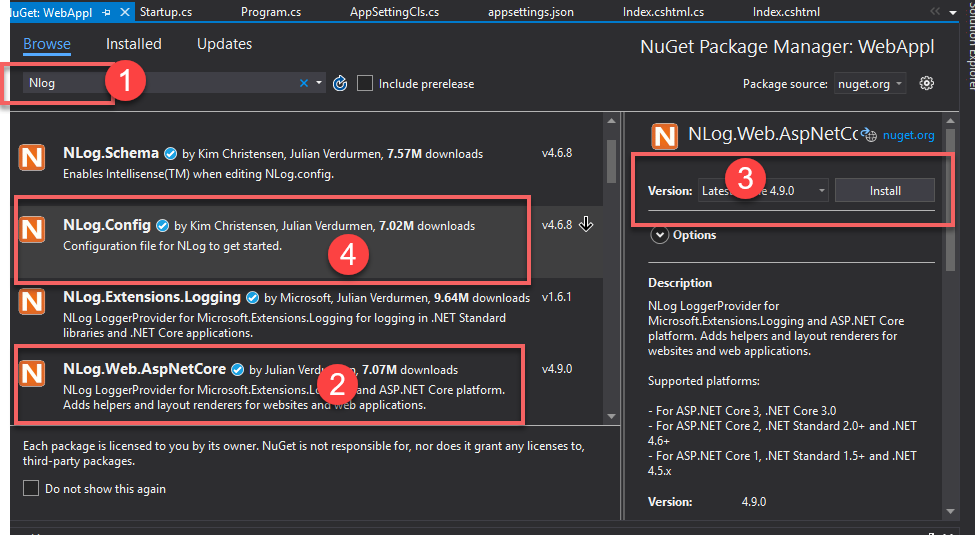
Log File

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

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 Refrences الى فى المشروع 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 اوتوماتيك و اخش اعدل فيه بقه براحتى بعد كده

NLog.Config و بما انى 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>

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

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