# 1、Core 的基本构造

不学习,跟咸鱼有什么区别?

既然如此,现在就开始疯狂干!

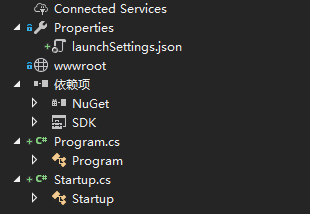
1,新建一个core 空项目



项目创建完后,你发现牛B的微软竟然还有入门的教程,虽然有部分还是英文,还没翻译好的,但是还是推荐认真读读.



然后在观察以下生成的目录信息



从上面的结构看出来,生成空的core web项目,连controller和views都没有,只包含了最核心的部分.

Properties——launchSettings.json 启动配置



1. Program, 程序的入口

看起来, 是不是跟控制台程序的入口一模一样,以前的web项目是寄宿在IIS下的,现在core web 项目是寄宿于控制台程序之中.

public class Program

{

public static void Main(string[] args)

{

var host = new WebHostBuilder()

.UseKestrel()//Kestrel服务器

.UseContentRoot(Directory.GetCurrentDirectory())//系统根目录

.UseIISIntegration()//IIS集成

.UseStartup<Startup>()//使用过owin的应该就不会陌生

.UseApplicationInsights()//这好像是一个新的,微软服务?

.Build();

host.Run();

}

}

Kestrel服务器：ASP.NET Core 托管模式并不直接监听请求；而是依赖于一个 HTTP server 实现来转发请求到应用程序。这个被转发的请求会以一组 feature 接口的形式被包装，然后被应用程序组合到一个 HttpContext中去。 ASP.NET Core 包含了一个托管的跨平台 web 服务器，被称为 Kestrel，它往往会被运行在一个如 IIS 或者 nginx 的生产 web 服务器之后。

1. Startup, asp.net core 的启动类,这个类的详细信息会在第二章具体分析.

public class Startup

{

public void ConfigureServices(IServiceCollection services)

{

}

public void Configure(IApplicationBuilder app, IHostingEnvironment env, ILoggerFactory loggerFactory)

{

loggerFactory.AddConsole();

if (env.IsDevelopment())

{

app.UseDeveloperExceptionPage();

}

app.Run(async (context) =>

{

await context.Response.WriteAsync("Hello World!");

});

}

}

IApplicationBuilder:构建应用程序的请求管道。

IHostingEnvironment:提供了当前的 EnvironmentName、WebRootPath 以及 Web 根文件提供者。

ILoggerFactory:提供了创建日志的机制。可以在Starup的构造函数里或者在Configure里使用内使用.

IServiceCollection:这里定义你使用的服务，譬如 identity，Entity Framework Core

1. wwwroot

项目根目录，从来存放静态资源，可以用 WebHostBuilder 来指定另外一个地址