**一、朝夕教育 Eleven**

Asp.Net框架，完成固定逻辑，把可变逻辑留下扩展点，满足特定需求。

基于事件驱动的框架（管道处理模型）

asp.net 框架提供了25个事件

【1.中间件】

扩展方法在core中随处可见

便于组件化

最小定义和扩展功能完全分开

startup

先注入，后addservice（先实例化，再执行方法）

Config

配置管道，http请求的处理过程

全部请求的处理过程

启动时执行，只一次

中断式middleware，请求都被该中间件处理了

app.Run(c=>c.Response.WriteAsync("Hello World!"));

管道处理模型

IApplicationBuilder

.New();

.Use(); //只有这一步是可以定制的

.Build();

可以通过Use去指定请求的处理过程

新版管道处理模型

俄罗斯套娃

看源码的方法】

中间件源码解读 1:20:00

【2. IOC】

ConfigureService 这里是core内置的ioc容器

IOC / DI 面向抽象编程

可以内置IOC可以单独拿出来在控制台试验

AddTransient 瞬时

多次provider.GetService，获取的不是同一个

AddSingleton 单例

多次是同一个

AddScoped 作用域单例，不是线程单例

只要是同一个provider，即使不同线程也是一个；

但如果不是一个privider，作用域就不同

【3.filter】

ExceptionFilterAttribute

三种注册方式

action

controller

全局

[ServiceFilter(typeof(CustomExceptionFilterAttribute))]

ActionFilter

AuthorizationFilter

ResultFilter

Filter执行顺序

Authorization

Resource

Exception

Action

Result

**二、软谋教育 Zailor**

【asp.net core】

1】

主机，负责web应用程序的启动和生存期管理

启动流程：

创建主机生成器-》配置主机-》创建主机-》运行

Host

host有两种：【】

默认主机

webhost

kestrel在linux下性能更高

但功能弱，需要配合反向代理

配置优先级：

命令行>应用配置>硬编码>环境变量

cmd启动方式

1.在proj目录dotnet run

2.在dll目录dotnet \*\*\*.dll

2】

配置web应用启动类：startup

配置web应用所需的服务和中间件

主机build后，run前

运行startup.ConfigureServices

IOC 控制反转

【反转？】

eg：吃东西，自己从冰箱拿 vs 妈妈给做好(依赖)喂到嘴里（注入）

目地：松耦合、可维护性、可测试性

IOC两个职责

注册类型

请求实例

利用IOC可以实现DI，依赖注入

可以看作升级版工厂，还负责物流配送，配送？

服务生命周期

singleton 单例 每次请求都使用相同的

transit 瞬时

scope 作用域 在同一个线程（请求）里，只实例化一次

【适用场景】

【注入时，为什么readonly,readonly什么时候生效】

# [C#基础知识系列八（const和readonly关键字）](https://www.cnblogs.com/aehyok/p/3529079.html)

const修饰的常量在声明的时候必须初始化;readonly修饰的常量则可以延迟到构造函数初始化

3】中间件

中间件职责：

决定是否将请求传递给管道中的下一个中间件

在管道中的下一个中间件的前后执行工作

管道短路，有必要，避免不必要工作

中间件、过滤器，都是基于AOP思想

区别：

过滤器关注如何实现功能

中间件是asp.net core的核心

布置管道=在管道里布置中间件

自定义中间件

约定：

具有类型为RequestDelegate的参数的公共方法

名为Invoke或者InvokeAsync的方法

asp.net core源码

第一步，执行批处理

没有sln，项目太大

直接打不开，执行startvs.cmd

源码，

Use暂时存放到IList，原始中间件

4】管道模型、配置、多环境、日志

var c1 = \_config["WesSetting:title"];1.基础方法，适合少量配置

2.绑定配置模型对象

全部绑定

部分绑定

3.注册配置选项的服务

自定义json文件

4.2】多环境

默认development

Startup

Startup\*\*\*

ConfigureServices

Configure\*\*\*Services

Configure

Configure\*\*\*

多配置文件？

# ASP.NET CORE 根据环境变量支持多个 appsettings.json

https://blog.csdn.net/sD7O95O/article/details/81463715

4.3】日志

内置日志简单

第三方NLog

4.4】路由

成对出现，中间可以插入其它中间件

app.UseRouting(); 负责匹配路由与终结点，解析出路由信息，写进HttpContext，传给下一个中间件

app.UseEndpoints

用来根据路由选择终点。

？？没有理解

【5】静态文件 部署

部署】

publish-edit-setting

依赖框架的部署FDD

dll文件，通用，可跨平台

部署包很小（依赖安装的runtime）

选择依赖框架-可移植

独立部署SCD

自带依赖，不依赖系统的core版本

包含一个可执行文件，win下为exe

选择self-contained

依赖框架的可执行文件FDE

依赖框架+平台