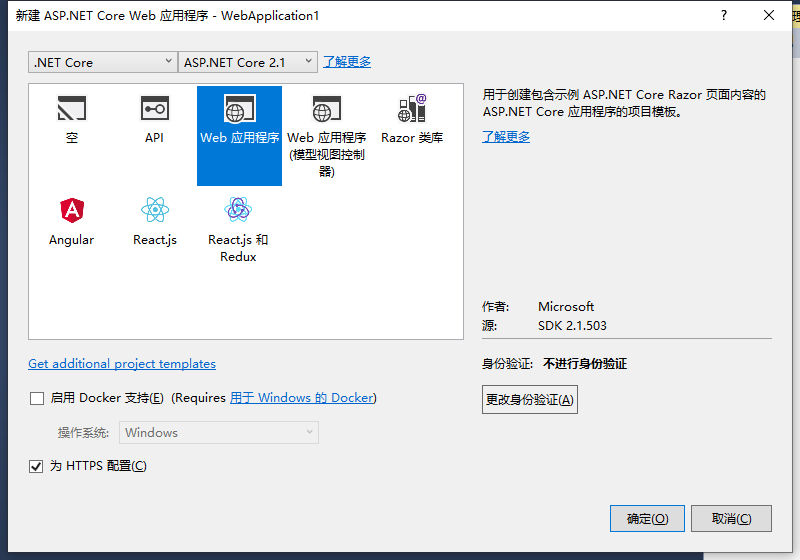
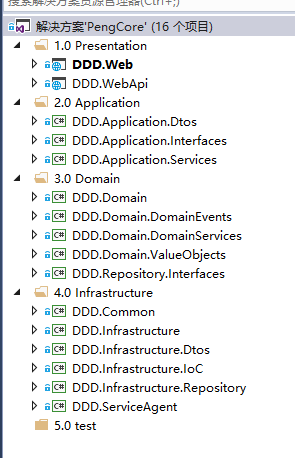
通用架构实现方案

## 创建web 应用程序



使用dotnet 的restore、build、run命令分别用来加载依赖，编译和运行

## 系统分层



### 1.0 Presentation 表现层

DDD.Web 表示层

用于提供用户的数据展现

AutoMapper

AutoMapper 和AutoMapper.Extensions.Microsoft.DependencyInjection

DDD.WebApi 数据接口层

用户提供数据接口

### 2.0 Application 应用层

DDD.Application.Dtos 应用层的数据对象传输

DDD.Application.Interfaces 应用层的数据接口

DDD.Application.Services 应用层的数据实现

### 3.0 Domain 领域层

DDD.Domain 领域实体数据

DDD.Domain.DomainEvents 领域事件

DDD.Domain.DomainServices 领域服务

DDD.Domain.ValueObjects 领域值对象

DDD.Repository.Interfaces 领域接口

定义接口基类和具体的业务接口类

### 4.0 Infrastructure 基础设施

DDD.Common 基础设施公共方法

DDD.Common 数据模型

DDD.Infrastructure.Dtos 基础设施数据对象传输

Appsettings.cs

引入：Microsoft.Extensions.Configuration 和 Microsoft.Extensions.Configuration.Json

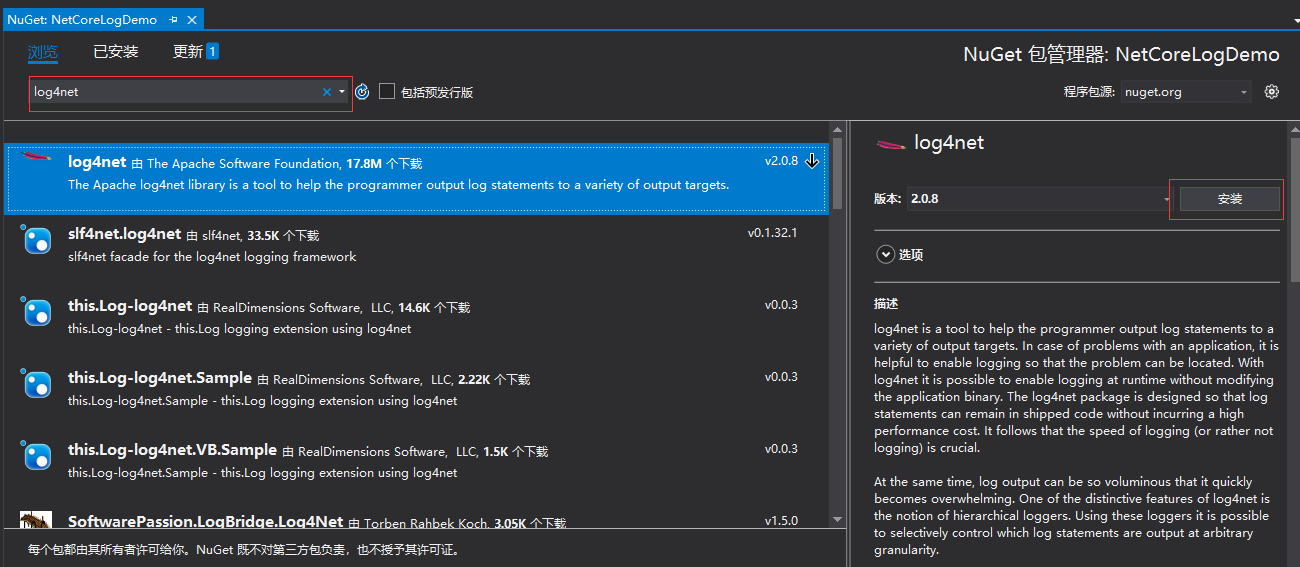
DDD.Infrastructure.IoC 控制反转

DDD.Infrastructure.Repository 基础设施方法实现

DDD.ServiceAgent 基础设施服务代理

## Log4net

1. 安装Log4net



1. 添加 log4net.config 到根目录（文件在项目中）
2. 在Startup.cs类里面配置使用log4net

public static ILoggerRepository repository { get; set; }

public Startup(IConfiguration configuration)

{

Configuration = configuration;

repository = LogManager.CreateRepository("NETCoreRepository");

// 指定配置文件

XmlConfigurator.Configure(repository, new FileInfo("log4net.config"));

}

四、在控制器里面注入log4net

public class HomeController : Controller

{

private ILog log;

public HomeController(IHostingEnvironment hostingEnv)

{

this.log = LogManager.GetLogger(Startup.repository.Name, typeof(HomeController));

}

public IActionResult Index()

{

log.Error("测试日志");

return View();

}

}