参考文章：https://www.cnblogs.com/artech/p/inside-asp-net-core-1.html

.net core 的依赖注入容器是 IServiceProvider

**IServiceProvider从何而来**

IServiceProvider由IServiceCollection的 BuildServiceProvider() 方法生成

.net core 默认提供的IServiceCollection为ServiceCollection

我们可以向IServiceCollection注入域，单例，临时3中服务类型

public static class ServiceCollectionExtensions

{

    public static IServiceCollection AddScoped<TService>(this IServiceCollection services) where TService: class;

    public static IServiceCollection AddSingleton<TService>(this IServiceCollection services) where TService: class;

    public static IServiceCollection AddTransient<TService>(this IServiceCollection services) where TService: class;

}

在站点生成前，我们可以向IServiceCollection注册我们的服务

站点生成时 .net core 会使用IServiceCollection生成IServiceProvider，而默认的IServiceProvider为ServiceProvider其不支持注册服务功能

**ServiceProvider如何提供域功能**

ServiceProvider通过ServiceScopeFactory提供域功能

internal class ServiceScopeFactory : IServiceScopeFactory

{

    private readonly ServiceProvider \_provider;

    public ServiceScopeFactory(ServiceProvider provider)

    {

        \_provider = provider;

    }

    public IServiceScope CreateScope()

    {

        return new ServiceScope(new ServiceProvider(\_provider));

    }

}

传给构造函数的ServiceProvider为根依赖注入容器

通过CreateScope创建的IServiceScope为某个域

public interface IServiceScope : IDisposable

{

    IServiceProvider ServiceProvider { get; }

}

域下的ServiceProvider属性则是该域的依赖注入容器

注：ServiceProvider中已注册一个ServiceScopeFactory的单例实例