IHostedService被注册到Host后，Host启动时，会允许所有注册的IHostedService

**注册IHostedService**

如下代码注册IHostedService

services.AddHostedService<MyHostedService>();

**IHostedService**

IHostedService示例如下：

    public class MyService : IHostedService

    {

        private IServiceProvider \_provider { get; }

        public MyService(IServiceProvider provider) {

            this.\_provider = provider;

        }

        public Task StartAsync(CancellationToken cancellationToken)

        {

            // HostedService运行

            return Task.CompletedTask;

        }

        public Task StopAsync(CancellationToken cancellationToken)

        {

            // HostedService停止

            return Task.CompletedTask;

        }

    }

**BackgroundService**

BackgroundService时一个抽象的后台服务类，其继承IHostedService，一般后台服务继承BackgroundService即可

    public class Worker : BackgroundService

    {

        private readonly ILogger<Worker> \_logger;

        public Worker(ILogger<Worker> logger)

        {

            \_logger = logger;

        }

        protected override async Task ExecuteAsync(CancellationToken stoppingToken)

        {

            while (!stoppingToken.IsCancellationRequested)

            {

                \_logger.LogInformation("Worker running at: {time}", DateTimeOffset.Now);

                await Task.Delay(1000, stoppingToken);

            }

        }

    }

**依赖注入域与IHostedService**

IHostedService依赖注入进来的IServiceProvider属于根IServiceProvider，如果想要转到域IServiceProvider，请使用如下方法

    public class MyService : IHostedService

    {

        private IServiceProvider \_provider { get; }

        public MyService(IServiceProvider provider) {

            this.\_provider = provider;

        }

        public Task StartAsync(CancellationToken cancellationToken)

        {

            // 创建域

            using (IServiceScope scope = \_provider.CreateScope())

            {

                MyClass1 myClass1 =

                    scope.ServiceProvider.GetRequiredService<MyClass1>();

            }

            return Task.CompletedTask;

        }

        public Task StopAsync(CancellationToken cancellationToken)

        {

            // HostedService停止

            return Task.CompletedTask;

        }

    }

**Windows服务**

使用IHostedService可以创建Windows服务，详情请看微软文档

<https://docs.microsoft.com/zh-cn/dotnet/core/extensions/windows-service>