**一个启动站点的例子**

public static void Main (string[] args) {

    new WebHostBuilder ()

        .UseKestrel ()

        .Configure (app => app.Run (async context => await context.Response.WriteAsync ("Hello World")))

        .Build ()

        .Run ();

}

**站点启动步骤解析**

上述步骤分解为

public static void Main (string[] args) {

    // WebHost生成器

    WebHostBuilder webHostBuilder = new WebHostBuilder ();

    // 配置Kestrel服务器

    webHostBuilder.UseKestrel ();

    // 配置中间件

    webHostBuilder.Configure(

        (IApplicationBuilder app) => app.Run(async context => await context.Response.WriteAsync("Hello World")));

    // 生成WebHost

    IWebHost webHost = webHostBuilder.Build ();

    // 运行WebHost

    webHost.Run ();

}

WebHostBuilder：是我们的站点生成器

UseKestrel()：配置Kestrel服务器，其实际上向WebHostBuilder配置了一个中间件

Configure()：配置中间件

Configure()不是注册中间件的动作，IApplicationBuilder.Run()才是注册中间件的动作

UseStartup<Startup>()：配置中间件

我们可以使用UseStartup代替Configure，其约定Startup必须有一个Configure，Configure方法的第一个参数必须是IApplicationBuilder app，其后参数则是通过依赖注入获取

public class Startup

{

    public void Configure (IApplicationBuilder app, IHostingEnvironment env) { }

}

Build()：利用注册的中间间，生成WebHost

Run()：启动WebHost，该方法调用Start方法启动WebHost