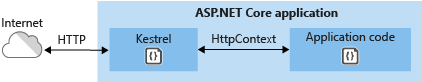
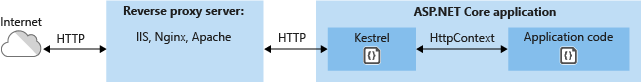
Kestrel 是一个跨平台的[适用于 ASP.NET Core 的 Web 服务器](https://docs.microsoft.com/zh-cn/aspnet/core/fundamentals/servers/index?view=aspnetcore-2.1)。

## **何时结合使用 Kestrel 和反向代理**

可以单独使用 Kestrel，也可以将其与反向代理服务器（如 IIS、Nginx 或 Apache）结合使用。 反向代理服务器接收到来自 Internet 的 HTTP 请求，并在进行一些初步处理后将这些请求转发到 Kestrel。





## **如何在 ASP.NET Core 应用中使用 Kestrel**

默认情况下，ASP.NET Core 项目模板使用 Kestrel。 在 Program.cs 中，模板代码调用 [CreateDefaultBuilder](https://docs.microsoft.com/zh-cn/dotnet/api/microsoft.aspnetcore.webhost.createdefaultbuilder)，后者在后台调用 [UseKestrel](https://docs.microsoft.com/zh-cn/dotnet/api/microsoft.aspnetcore.hosting.webhostbuilderkestrelextensions.usekestrel)。

public static void Main (string[] args) {

    CreateWebHostBuilder (args).Build ().Run ();

}

public static IWebHostBuilder CreateWebHostBuilder (string[] args) =>

    WebHost

    .CreateDefaultBuilder (args)

    .UseStartup<Startup> ();

### **Kestrel 选项**

.UseKestrel(options =>

{

    options.Limits.MaxConcurrentConnections = 100;

    options.Limits.MaxConcurrentUpgradedConnections = 100;

    options.Limits.MaxRequestBodySize = 10 \* 1024;

    options.Limits.MinRequestBodyDataRate =

        new MinDataRate(bytesPerSecond: 100, gracePeriod: TimeSpan.FromSeconds(10));

    options.Limits.MinResponseDataRate =

        new MinDataRate(bytesPerSecond: 100, gracePeriod: TimeSpan.FromSeconds(10));

    options.Listen(IPAddress.Loopback, 5000);

    options.Listen(IPAddress.Loopback, 5001, listenOptions =>

    {

        listenOptions.UseHttps("testCert.pfx", "testPassword");

    });

});

### **终结点配置**

默认情况下，ASP.NET Core 绑定到：

* http://localhost:5000
* https://localhost:5001（存在本地开发证书时）