

# ASP.NET Core 基础系列 (2) (托管模型)

原创 痕迹 CodeShare 1周前

收录于话题

#ASP.NET Core

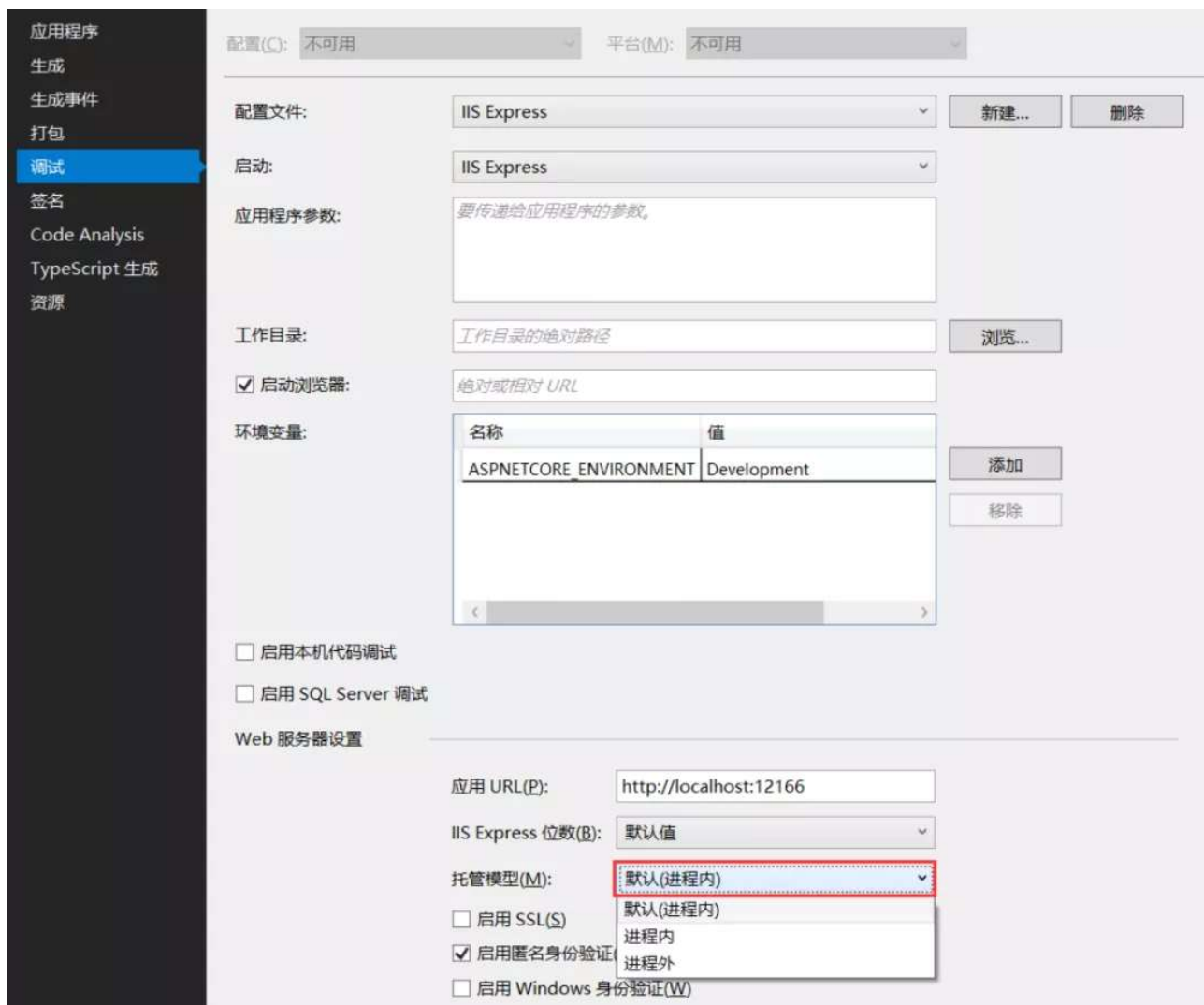
8个

这一章我们来讨论一下 `CreateDefaultBuilder()` 方法究竟对配置和设置 Web 服务器的作用。从托管的角度来看，一个 ASP.NET Web 应用程序可以通过两种方式托管，即进程托管(InProcess)或进程外托管(OutOfProcess)。

**注：** 当我们使用任何模板创建新 ASP.NET Core Web 应用程序时，默认情况下，使用 InProcess 托管创建项目文件，该托管用于在 IIS 或 IIS Express 方案中托管应用程序。

## 如何验证是否在进程内？

为了验证上面的观点，打开项目属性。右键单击项目，然后从上下文菜单中选择属性选项。打开属性窗口后，选择"调试"并查看"托管模型"下拉列表的值，如下图所示。下拉列表包含三个值，即默认值（进程内）、进程内和进程外。



那么, 确认默认情况下它使用"进程中"托管模型。

## 如何在ASP.NET Core中配置InProcess托管?

要为 ASP.NET Core Web 应用程序配置 InProcess 托管, 只有一个简单的设置, 只需将 元素添加到值为 InProcess 的应用程序项目文件中。为此, 只需从解决方案资源管理器右键单击应用程序, 然后单击上下文 菜单中的 "编辑项目文件" 选项, 如下代码所示。

```
<Project Sdk="Microsoft.NET.Sdk.Web">

  <PropertyGroup>
    <TargetFramework>net5.0</TargetFramework>
    <AspNetCoreHostingModel>InProcess</AspNetCoreHostingModel>
  </PropertyGroup>

</Project>
```

打开应用程序项目文件后, 请对其进行修改, 如下所示。此处添加元素并将其值设置为 InProcess。该节点也可以是OutOfProcess。

将托管模型设置为 InProcess 时会发生什么?

在 InProcess 托管的情况下（即当 `CreateDefaultBuilder()` 将值视为项目文件中 `AspNetCoreHostingModel` 元素的 InProcess 时），场景后面的 `CreateDefaultBuilder()` 方法内部调用 `UseIIS()` 方法。

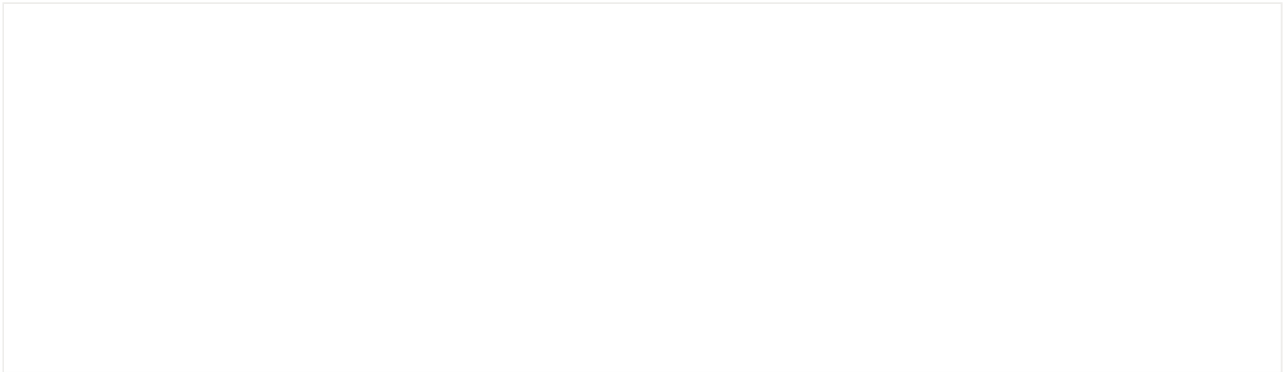
然后在 IIS 工作程序（即 IIS 的 `w3wp.exe` 和 `IISExpress.exe` 的 `IISExpress`）中托管应用程序。

从性能的角度来看，InProcess 托管模型提供的请求吞吐量明显高于 OutOfProcess 托管模型。

在 IIS 中，将用于执行应用程序的进程名称是 `w3wp`。同样，如果是 IIS Express，则进程名称将是 `iisexpress`。

### 如何确定是否在 InProcess 进程下执行？

默认情况下，如果您要运行该应用程序，那么您将在浏览器中看到以下 “Hello World!” 输出。

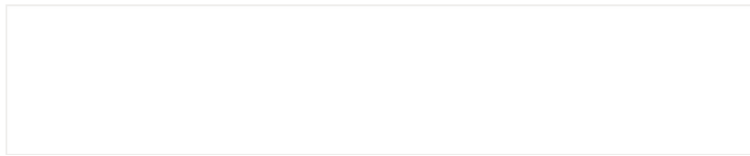


下面，标记了输出 “Hello World” 位置的代码片段，位于 `Startup` 类的 `Configure()` 方法中，如下所示。

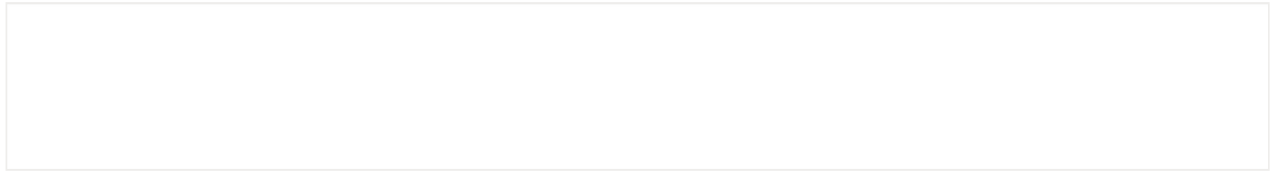
为了确认该应用程序是在InProcess进程下执行, 将"Hello World!" 修改为显示当前的进程名称, 通过调用:

`System.Diagnostics.Process.GetCurrentProcess().ProcessName`获取当前的进程名称, 如下所示。

再次启动程序, 将在浏览器中显示以下信息, 如下所示。



这是因为默认情况下，Visual Studio 在运行应用程序时使用 IIS Express，如下图所示。



## 什么是 IIS Express?

IIS Express 是 IIS 的轻量级自包含版本。它针对 Web 应用程序开发进行了优化。您需要记住的最重要的一点是，我们仅在开发中使用 IIS Express，而不是生产。在生产中，我们通常使用 IIS。

## 为什么 InProcess 比 OutOfProcess 提供更好的性能?

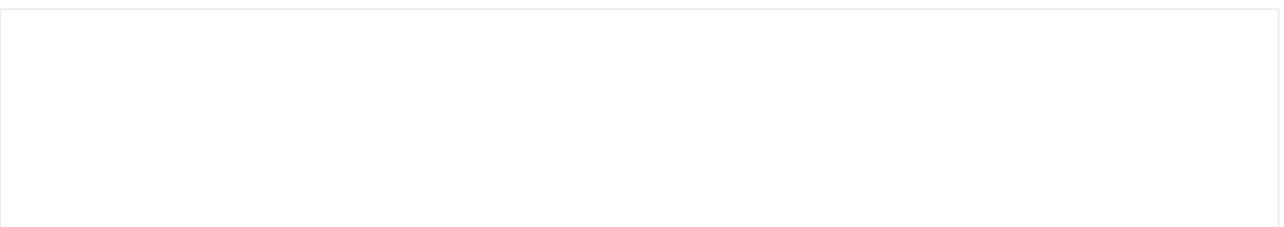
在进程外 (OutOfProcess) 托管的情况下，有 2 台 Web 服务器

1. 一个内部 Web 服务器
2. 一个外部 Web 服务器。

内部 Web 服务器称为 Kestrel，外部 Web 服务器可以是 IIS、Nginx\*\*\*\* 或 Apache。使用 InProcess 托管模型时，只有一个 Web 服务器，即 IIS。因此，对于 InProcess 托管模型，我们在内部和外部 Web 服务器之间导航请求没有性能损失。这就是 InProcess 托管模型提供比 OutOfProcess 托管模型更高的请求吞吐量的原因。

## InProcess Hosting(进程托管)

在上面的内容讲到，我们可以通过右键选择属性，更改应用程序的托管模型，也可以通过右键编辑项目文件编辑节点来设置，如下所示。



在 ASP.NET Core 中，使用 Inprocess 托管模型，我们的应用程序将在 IIS 工作程序中托管。最重要的一点，需要记住的是，我们只有一个 Web 服务器，即 IIS 服务器的情况下，InProcess 托管，这将托管我们的应用程序，如下图所示。

## 什么是Kestrel Web 服务器？

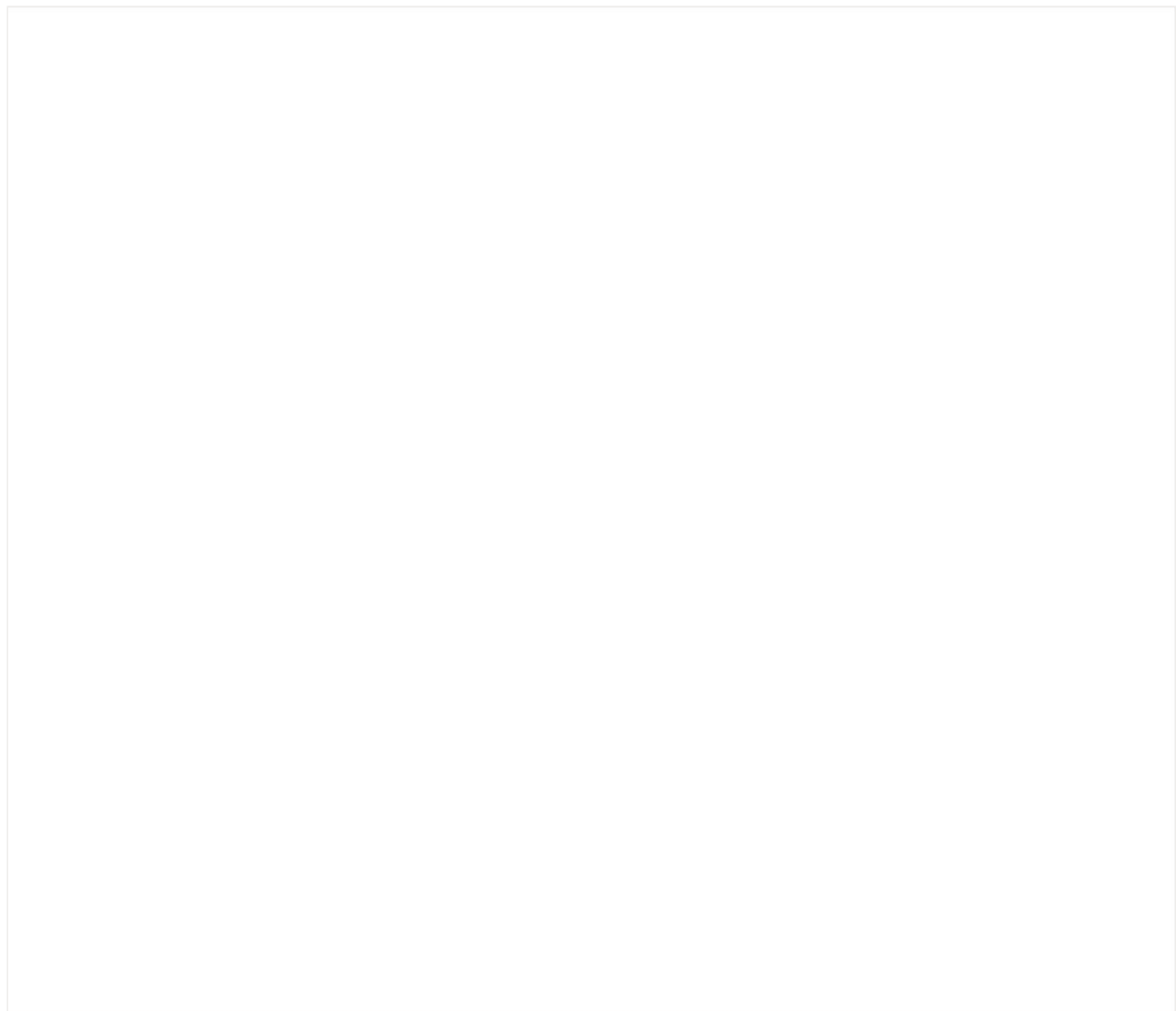
ASP.NET Core是一个跨平台框架。这意味着它支持在不同类型的操作系统（例如Windows，Linux或Mac）上开发和运行应用程序。

Kestrel是ASP.NET Core应用程序的跨平台Web服务器。这意味着该服务器支持ASP.NET Core支持的所有平台和版本。默认情况下，它包含在.NET Core应用程序中作为内部Web服务器。

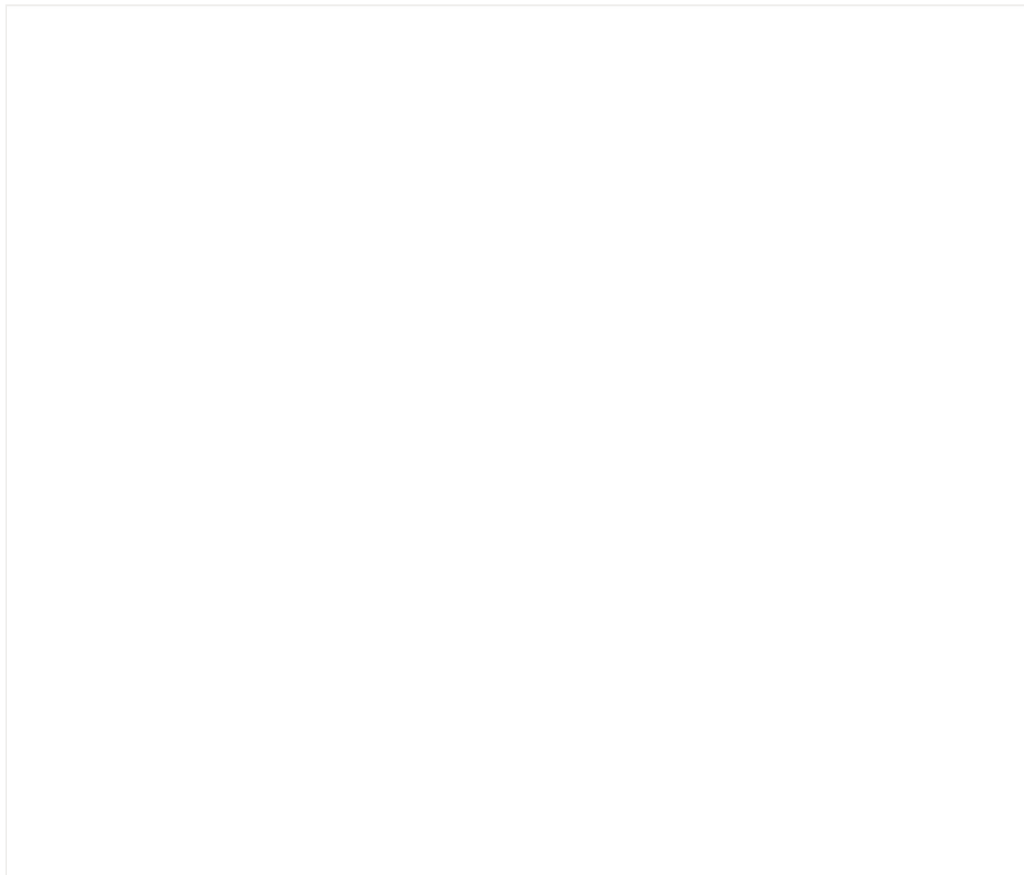
Kestrel Web服务器通常用作边缘服务器，即面向Internet的Web服务器，它直接处理来自客户端的传入HTTP请求。对于Kestrel Web服务器，用于承载和运行ASP.NET Core应用程序的进程名称是项目名称。

## 如何使用 Kestrel Web 服务器运行应用程序？

在使用 Kestrel 服务器运行应用程序之前，让我们首先打开应用程序的属性文件夹中的launchSettings.json 文件。打开launchSettings.json 文件后，默认情况下，您将找到以下代码。



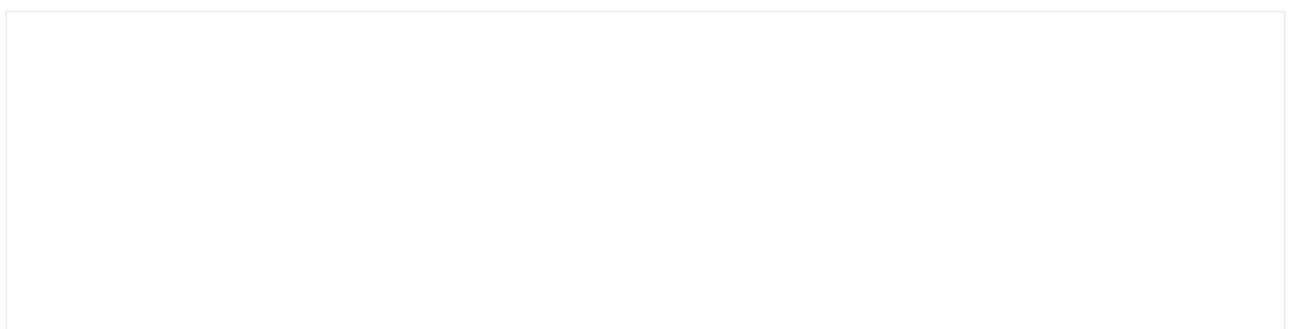
在这里，您可以看到，我们有两个部分。一种用于IIS Express（IIS服务器），另一种用于Kestrel服务器。在Visual Studio中，您可以找到上述两个配置文件（IIS Express和FirstCoreWebApplication），如下所示。



如果选择 IIS Express，则它将使用 IIS 服务器，如果您选择 FirstCoreWeb 应用程序，则它将使用 Kestrel 服务器。

使用 IIS Express 运行应用程序：

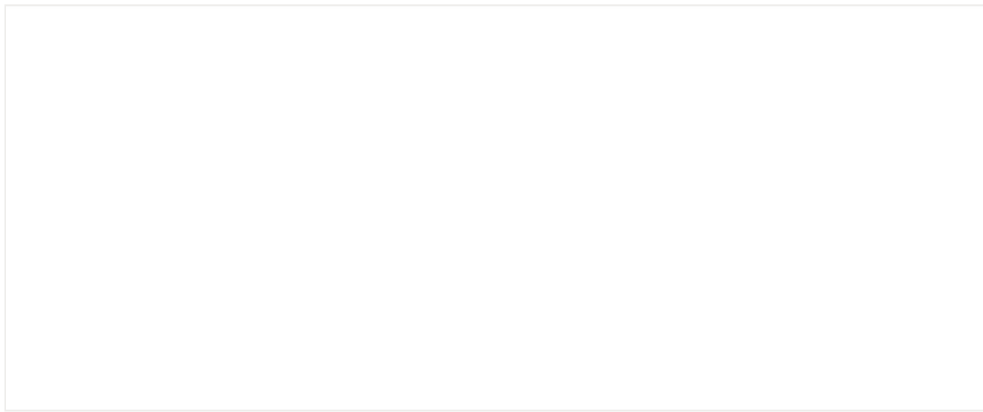
如果您使用 IIS Express 运行应用程序，则它将使用启动设置.json 文件中提及的 URL 和端口号。若要证明这使用 IIS Express 运行应用程序，请参阅如下所示的输出(左边是配置文件、右边是浏览器输出)。



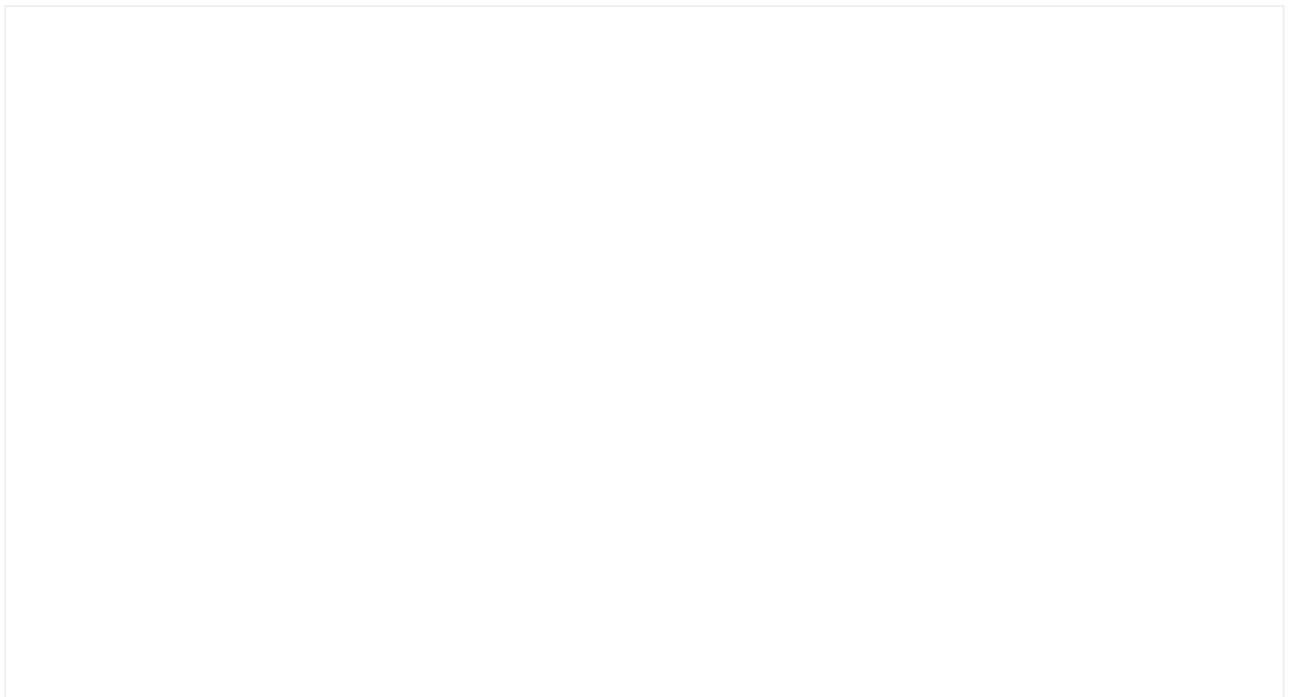
使用 Kestrel 服务器运行应用程序：

为了使用 Kestrel 服务器在 Visual Studio 中运行应用程序，首先，您需要修改你的启动方式为你当前的应用程序名称，如下所示。





选择 应用程序方式启动后，现在运行该应用程序。在这里，我们需要观察两件事。首先，它将启动命令提示符并使用 Kestrel 服务器托管应用程序，如下所示。在这里，您需要关注 URL 和端口号，并且应该是launchSettings.json文件的 WebApplication4 (这里根据您的程序名称显示) 配置文件中提到的URL和端口号。请参阅如下所示的输出(左边是配置文件、右边是浏览器输出以及打开的.NET CLI输出窗口)。



**注：** 在这个的示例中，对于 IIS Express，端口号为 12166，辅助进程为 iisexpress，而对于 Kestrel 服务器，端口号为 5000，辅助进程名称为 WebApplication4 应用程序（它只不过是您的应用程序名称）。

[阅读原文](#) 阅读 104 文章已于2021/03/09修改

分享

收藏

赞 5

在看 1