

ASP.NET Core 基础系列（3）（启动设置）

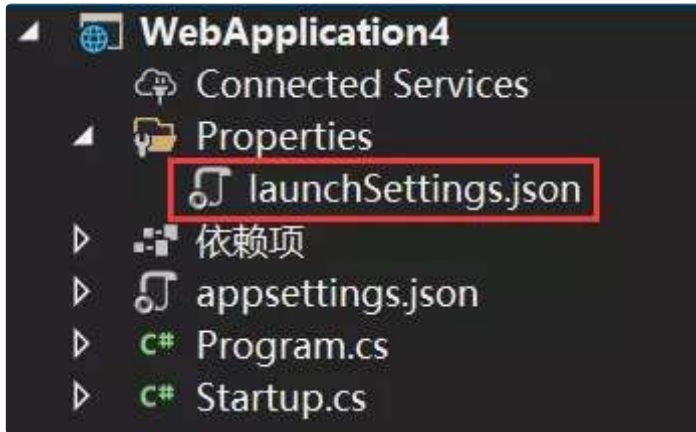
原创 痕迹 CodeShare 1周前

收录于话题

#ASP.NET Core

8个

这一章,我们了解一下launchSettings.json的作用。

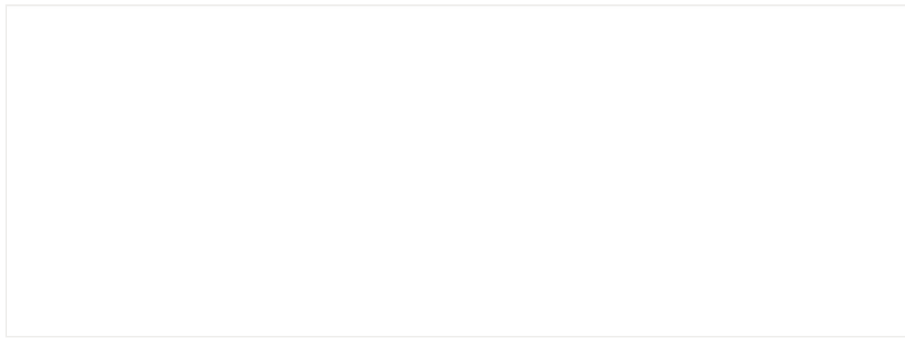


打开 launchSettings.json 文件后，默认情况下，您将找到以下代码。

```
{
  "iisSettings": {
    "windowsAuthentication": false,
    "anonymousAuthentication": true,
    "iisExpress": {
      "applicationUrl": "http://localhost:1118",
      "sslPort": 0
    }
  },
  "profiles": {
    "IIS Express": {
      "commandName": "IISExpress",
      "launchBrowser": true,
      "environmentVariables": {
        "ASPNETCORE_ENVIRONMENT": "Development"
      }
    },
    "WebApplication4": {
      "commandName": "Project",
      "dotnetRunMessages": "true",
      "launchBrowser": true,
      "applicationUrl": "http://localhost:5000",
      "environmentVariables": {
        "ASPNETCORE_ENVIRONMENT": "Development"
      }
    }
  }
}
```

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WebApplication4），如下所示。



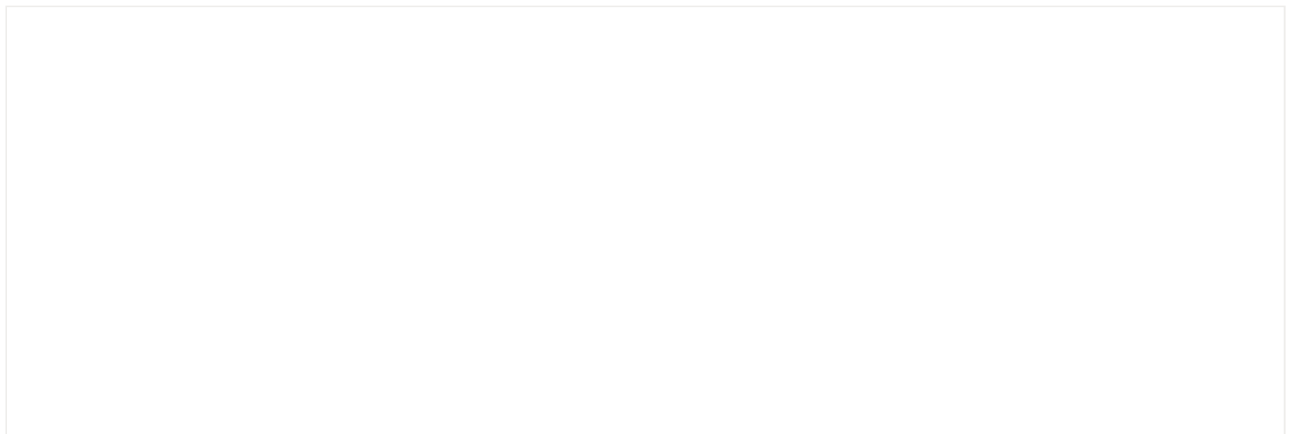
如果选择 IIS Express, 则它将使用 IIS 服务器。

如果您选择 WebApplication4 应用程序, 则它将使用 Kestrel 服务器。

launchSettings.json文件的 commandName 属性的值可以是以下任何一个:

1. IISExpress
2. IIS
3. Project

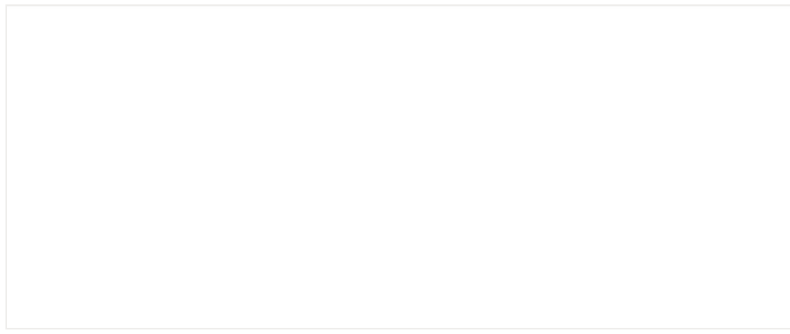
launchSettings.json文件的**commandName**属性值以及应用程序项目文件中的AspNetCoreHostingModel元素值将确定将要使用和处理传入HTTP请求的内部和外部Web服务器（反向代理服务器）。为了更好地理解, 请看下图。



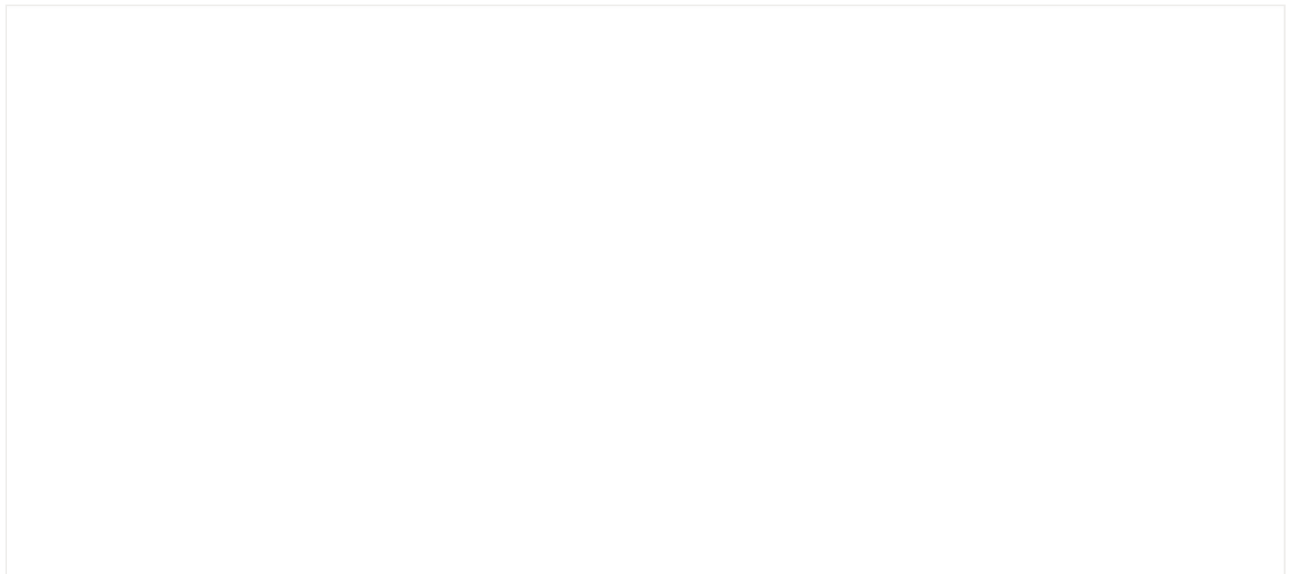
如何验证其工作模式?

方法1:

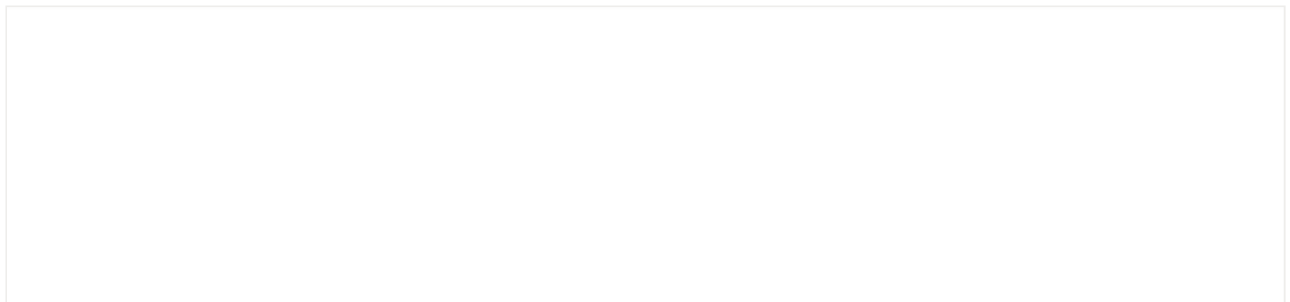
当我们在**launchSettings.json**文件中将**commandName**用作Project时, ASP.NET Core将忽略AspNetCoreHostingModel值。Kestrel是唯一将托管应用程序并处理传入请求的服务器。让我们证明这一点。现在,我们需要将启动Profile设置为WebApplication4, 如下所示。



如果查看launchSettings.json文件, 那么您将看到WebApplication4配置文件使用"commandName": "Project"值, 并且请注意下面所示的应用程序URL。在我的应用程序中, URL为http://localhost:5000, 端口号在您的示例中可能有所不同。



现在, 将应用程序项目文件中的AspNetCoreHostingModel元素值更改为InProcess, 如下所示。



然后, 启动该项目, 可以看到首先启动了CMD,在这种情况下将使用kestrel服务器托管应用程序。

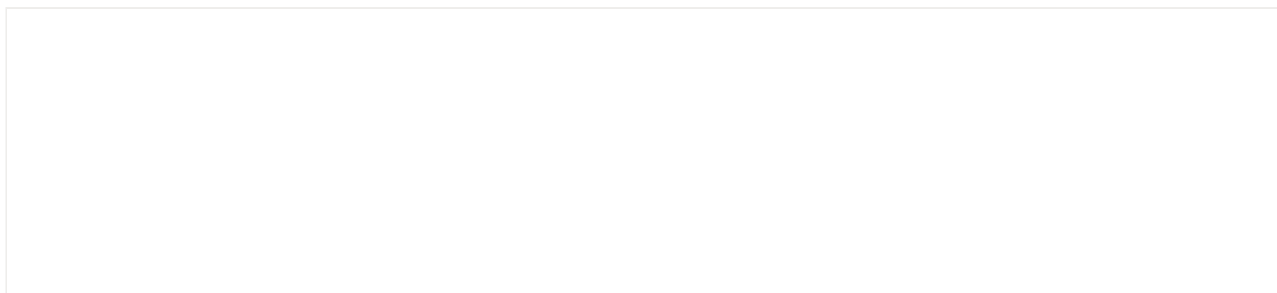
注: 因为当CommandName值为Project时, 它将忽略AspNetCoreHostingModel值, 而Kestrel是唯一将托管和处理传入请求的服务器。

方法2:

如果我们使用命令名作为 IISExpress 配置文件, 如果我们将 AspNetCoreHostingModel 值设置为 OutOfProcess, 则ASP.NET Core使用IIS Express作为外部 Web 服务器, Kestrel 是内部 Web 服务器。外部 Web 服务器 (即

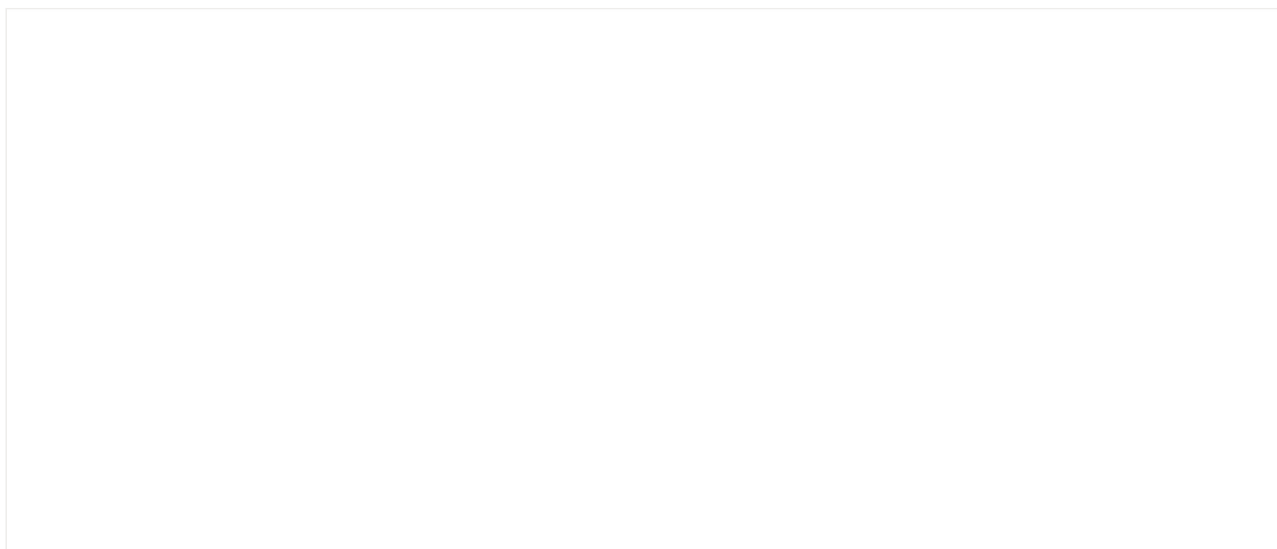
IIS Express) 将接收传入的 HTTP 请求, 然后将请求转发到内部 Web 服务器, 即将处理请求的 Kestrel。让我们证明这一点。

由于我们已经将启动配置文件设置为 IIS Express, 我们只需要将 `AspNetCoreHostingModel` 元素值更改为应用程序的项目文件中的 `OutOfProcess`, 如下所示。



运行应用程序, 浏览器输出了项目名称, 如下所示, 因为请求由 Kestrel Web Server 服务器处理。

注: 通过运行时, 输出当前进程名称即可。



[阅读原文](#) 阅读 82

分享

收藏

赞 2

在看