.NET 开发经典名著

# ASP.NET Core 3 高级编程(第 8 版)

(上册)

[英] 亚当·弗里曼(Adam Freeman) 著

杜静芬 程凤娟 译

清华大学出版社

北 京

北京市版权局著作权合同登记号　图字：01-2021-3335
EISBN：978-1-4842-5439-4
First published in English under the title
Pro ASP.NET Core 3: Develop Cloud-Ready Web Applications Using MVC, Blazor, and Razor Pages, Eighth Edition
By Adam Freeman


**图书在版编目(CIP)数据**

ASP.NET Core 3高级编程：第8版 / (英)亚当 • 弗里曼 (Adam Freeman) 著；杜静芬，程凤娟译. — 北京：清华大学出版社，2021.6
(.NET 开发经典名著)
书名原文：Pro ASP.NET Core 3: Develop Cloud-Ready Web Applications Using MVC, Blazor, and Razor Pages, Eighth Edition
ISBN 978-7-302-58271-7

Ⅰ. ①A…　Ⅱ. ①亚… ②杜… ③程…　Ⅲ. 网页制作工具－程序设计　Ⅳ. ①TP393.092.2

中国版本图书馆 CIP 数据核字(2021)第 101201 号

**责任编辑**：王　军　韩宏志
**装帧设计**：孔祥峰
**责任校对**：成凤进
**责任印制**：宋　林

**出版发行**：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
**印 装 者**：三河市国英印务有限公司
**经　　销**：全国新华书店
**开　　本**：170mm×240mm　　**印　　张**：73　　**字　　数**：2011 千字
**版　　次**：2021 年 7 月第 1 版　　**印　　次**：2021 年 7 月第 1 次印刷
**定　　价**：268.00 元(全二册)

产品编号：089796-01

# 译 者 序

ASP.NET Core 是一款由微软创建的，用于构建 Web 应用、API、微服务的 Web 框架。它使用常见的模式，如 MVC(Model-View-Controller)、依赖注入和一个由中间件构成的请求处理管道。ASP.NET Core 是重新设计的 ASP.NET 4.x，更改了体系结构，形成了更精简的模块化框架。

ASP.NET Core 运行在微软的.NET 运行时库上，有几种语言(C#、Visual Basic 和 F#)可用来编写 ASP.NET Core 程序。C#是最常见的选择，可在 Windows、macOS 和 Linux 上构建并运行 ASP.NET Core 应用。

为什么要用 ASP.NET Core 开发应用程序？现存的 Web 框架选项已经很多了，如 Node/Express、Spring、Ruby on Rails、Django、Laravel 等，数不胜数。ASP.NET Core 具有如下优点。

- 快速：因为.NET Core 是编译运行的，执行速度远高于解释执行的语言，如 JavaScript 或 Ruby，ASP.NET Core 也已为多线程和异步任务做了专门优化。与使用 Node.js 写的代码相比，执行速度高出 5～10 倍是很正常的。
- 生态：在 NuGet(.NET 的包管理系统，如 NPM、RubyGems 或 Maven)中有成千上万的软件包。有现成的包可用来完成 JSON 反序列化、数据库连接、PDF 生成等。
- 安全：通过 ASP.NET Core，开发者可轻松配置和管理应用的安全性。ASP.NET Core 的功能包括管理身份验证、授权、数据保护、HTTPS 强制、应用机密、请求防伪保护及 CORS 管理。 通过这些安全功能，可生成安全可靠的 ASP.NET Core 应用。
- 跨平台：能在 Windows、macOS 和 Linux 上开发和运行。
- 开源：出现问题时，可阅读其源代码，来获取解决问题的方法。

本书深入浅出地介绍 ASP.NET Core 基础及实战等方面的知识，共分 4 个部分。第Ⅰ部分介绍 ASP.NET Core。除了设置开发环境和创建第一个应用程序外，还介绍对 ASP.NET Core 开发最重要的 C#特性和如何使用 ASP.NET Core 开发工具。第Ⅱ部分描述 ASP.NET Core 平台的主要特性，解释如何处理 HTTP 请求，如何创建和使用中间件组件，如何创建路由，如何定义和使用服务，以及如何与 Entity Framework Core 一起工作。第Ⅲ部分解释如何创建不同类型的应用程序，包括 RESTful Web 服务以及使用控制器和 Razor Pages 的 HTML 应用程序。第Ⅳ部分解释如何使用 Blazor 服务器创建应用程序，如何使用实验性的 Blazor WebAssembly，以及如何使用 ASP.NET Core 验证用户身份和授予访问权限。

本书对于任何一名 C#开发者来说都是一本宝贵的指导书。书中所涉及的对于代码设计的实用建议是非常宝贵的。高效率的.NET 开发者需要对他所选择的语言有很深的理解。本书作者以令人惊叹的能力，把极其复杂的问题拆解为可消化的、易理解的一个个小问题，进行合理探讨，以一定的洞察力将知识传授给读者，并教给读者如何书写实践性强、干净简单且更容易理解的代码。无论是 C#新手还是资深开发者，都能通过阅读本书而有所收获。

本书文字简洁明快、流畅，既适合初学者及具有.NET 基础的开发者阅读，还可作为大中专院校计算机、通信、电子信息、自动化等相关专业的教材；也可供软件项目管理人员、开发团队成员学习参考。

这里要感谢清华大学出版社的编辑，他们为本书的翻译投入了巨大热情并付出了很多心血。没有他们的帮助和鼓励，本书不可能顺利付梓。

对于这本经典之作，译者本着“诚惶诚恐”的态度，在翻译过程中力求“信、达、雅”，但鉴于译者水平有限，失误在所难免，如果你有任何意见和建议，欢迎指正。

# 作者简介

Adam Freeman 是一位经验丰富的 IT 专业人士，曾在多家公司担任高级职位，后担任一家全球银行的首席技术官和首席运营官。现在退休了，他把时间花在写作和长跑上。

# 技术审校者简介

Fabio Claudio Ferracchiati 是一名使用微软技术的高级顾问和高级分析师/开发人员。他为 BluArancio (www.bluarancio.com)工作。他是一名.NET 解决方案开发人员和应用程序开发人员，也是一名多产的作者和技术评论员。在过去十年里，他为意大利和国际杂志撰写文章，并与他人合著了十多本关于各种计算机主题的书籍。

# 目　录

# 第 I 部分

# 介绍 ASP.NET Core

- 第 1 章　ASP.NET Core 上下文
- 第 2 章　入门
- 第 3 章　第一个 ASP.NET Core 应用程序
- 第 4 章　使用开发工具
- 第 5 章　C#的基本特点
- 第 6 章　测试 ASP.NET Core 应用程序
- 第 7 章　SportsStore：一个真正的应用程序
- 第 8 章　SportsStore：导航和购物车
- 第 9 章　SportsStore：完成购物车
- 第 10 章　SportsStore：管理
- 第 11 章　SportsStore：安全与部署

# 第 1 章

# ASP.NET Core 上下文

## 1.1 了解 ASP.NET Core

ASP.NET Core 是微软的 Web 开发平台。原来的 ASP.NET 是在 2002 年引入的，它经过了几次再创造和转世才成为 ASP.NET Core 3，这是本书的主题。

ASP.NET Core 由一个处理 HTTP 请求的平台、一系列用于创建应用程序的主要框架和提供支持特性的次要实用程序框架组成，如图 1-1 所示。

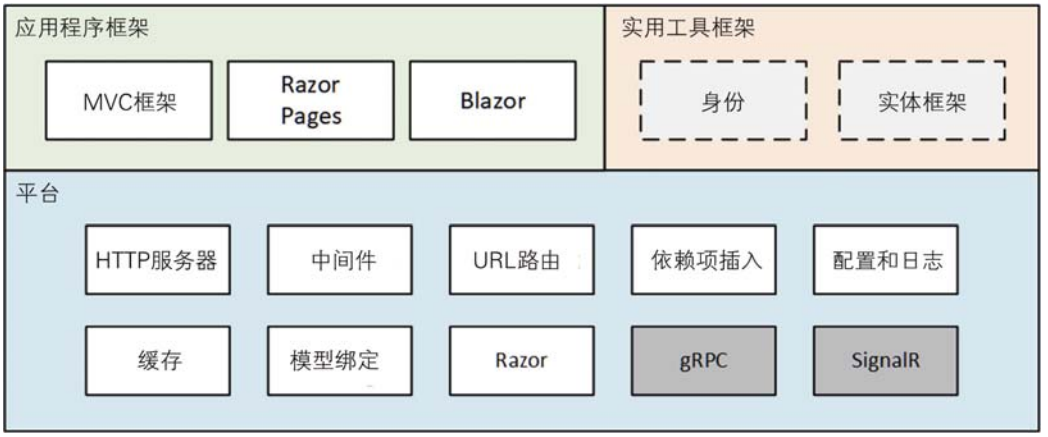


图 1-1　ASP.NET Core 的结构

### 1.1.1 理解应用程序框架

开始使用 ASP.NET Core，发现有不同的应用框架可用时，可能会感到困惑。其实，这些框架是互补的，可以解决不同的问题，或者，对于某些特性，可采用不同方式解决相同的问题。理解这些框架之间的关系意味着理解 Microsoft 所支持的不断变化的设计模式，如下面各节所述。

#### 1. 理解 MVC 框架

MVC 框架是在 ASP.NET Core 以前的时代引入的。原来的 ASP.NET 依赖于一种称为 Web Pages 的开发模型，这种模型重新创建了编写桌面应用程序的体验，但导致了无法很好扩展的、笨拙的

Web 项目。MVC 框架与 Web 页面一起引入，开发模型包含 HTTP 和 HTML 的特性，而不是试图隐藏它们。

MVC 代表 Model-View-Controller，是一种描述应用程序形状的设计模式。MVC 模式强调关注点分离，其中功能区域是独立定义的，这是对 Web 页面导致的模糊架构的有效解决方案。

早期版本的 MVC 框架建立在 ASP.NET 基础之上，最初是为 Web 页面设计的，这导致了一些笨拙的特性和解决方案。随着向.NET Core 的迁移，ASP.NET 成为 ASP.NET Core，MVC 框架是在一个开放的、可扩展的、跨平台的基础上重新构建的。

MVC 框架仍然是 ASP.NET Core 的重要组成部分。但是，随着单页应用程序(SPA)的兴起，它通常的使用方式已经发生了变化。在 SPA 中，浏览器发出一个单独的 HTTP 请求，然后接收一个 HTML 文档，该文档将提供一个富客户端，通常是用 JavaScript 客户端编写的，比如 Angular 或 React。SPA 的转变意味着 MVC 框架最初追求的清晰分离并不重要，强调遵循 MVC 模式不再是必要的，但 MVC 框架仍然是有用的(用于通过 Web 服务支持 SPA，详见第 19 章)。

### 2. 把模式放在应有的位置上

模式只是其他人在其他项目中遇到的问题的解决方案。如果自己也面临同样的问题，了解以前是如何解决的会很有帮助。但这并不意味着必须完全遵循这个模式，或者根本不遵循，只要明白后果就行。例如，如果模式的目的是使项目可管理，而选择背离该模式，就必须接受项目可能更难管理这一事实。但是盲目地遵循模式可能比没有模式更糟糕，而且没有模式适合每个项目。

建议自由地使用模式，根据需要调整它们，忽略那些混淆模式和戒律的狂热分子。

### 3. 理解 Razor Pages

MVC 框架的一个缺点是，在应用程序开始生成内容之前，它可能需要进行大量的准备工作。尽管存在结构上的问题，但 Web 页面的一个优点是可以在几小时内创建简单的应用程序。

Razor Pages 采用了 Web 页面的开发风格，并使用最初为 MVC 框架开发的平台特性来实现它。代码和内容混合在一起，形成自包含的页面；这重新提高了 Web 页面的开发速度，而不存在一些潜在的技术问题(尽管扩大复杂项目的规模仍然是个问题)。

Razor Pages 可以与 MVC 框架一起使用，这也是笔者倾向于使用的方式。笔者使用 MVC 框架编写应用程序的主要部分，使用 Razor Pages 编写次要功能，比如管理和报告工具。第 7～11 章就使用了这种方法，那几章开发了一个现实的 ASP.NET Core 应用程序，名为 SportsStore。

### 4. 理解 Blazor

JavaScript 客户端框架的兴起对 C#开发人员来说是一个障碍，他们必须学习一种不同的、有些特殊的编程语言。笔者已经爱上 JavaScript，它和 C#一样流畅、富有表现力。但是，精通一种新的编程语言需要时间和投入，尤其是与 C#有根本区别的语言。

Blazor 试图通过允许将 C#用于编写客户端应用程序来弥补这一差距。Blazor 有两个版本：Blazor Server 和 Blazor WebAssembly。Blazor Server 是 ASP.NET Core 中稳定且受支持的部分。它通过使用到 ASP.NET Core 的持久 HTTP 连接工作。应用程序的 C#代码在这里执行。Blazor WebAssembly 是一个实验性版本，它更进一步，在浏览器中执行应用程序的 C#代码。如第 33 章所述，Blazor 的两种版本都不适合所有的情况，但它们都为 ASP.NET Core 开发的未来指明了方向。

### 1.1.2 理解实用程序框架

有两个框架与 ASP.NET Core 密切相关，但不直接用于生成 HTML 内容或数据。Entity Framework Core 是 Microsoft 的对象-关系映射(ORM)框架，它将存储在关系数据库中的数据表示为.NET 对象。Entity Framework Core 可以在任何.NET Core 应用程序中使用，它通常用于访问 ASP.NET Core 应用程序中的数据库。

ASP.NET Core Identity 是 Microsoft 的认证和授权框架，用于验证 ASP.NET Core 应用程序中的用户凭证，限制对应用程序特性的访问。

本书只描述了这两个框架的基本特性，重点是大多数 ASP.NET Core 应用程序所需要的功能，但它们都是复杂的框架，太大了，无法在一本关于 ASP.NET Core 的书中详细描述。

**未来版本的主题**

本书没有足够的篇幅介绍每一个 Entity Framework Core 和 ASP.NET Core Identity 特性，而只关注大多数项目都需要的方面。如果你认为某主题应该包括在本书的下一版或进行深度解析，请把建议发送到 adam@adam-freeman.com。

### 1.1.3 了解 ASP.NET Core 平台

ASP.NET Core 平台包含接收和处理 HTTP 请求以及创建响应所需的底层特性。有一个集成的 HTTP 服务器、一个处理请求的中间件组件系统和应用程序框架所依赖的核心特性，如 URL 路由和 Razor 视图引擎。

大部分开发时间都花费在应用程序框架上，但为了有效使用 ASP.NET Core，需要理解该平台提供的强大功能；没有这些功能，高层框架就无法运行。本书第 II 部分详细介绍 ASP.NET Core 平台的工作原理，解释它提供的特性是如何支撑 ASP.NET Core 开发的各个方面的。

本书没有描述两个值得注意的平台特性：SignalR 和 gRPC。SignalR 用于在应用程序之间创建低延迟通信通道，为本书第Ⅳ部分中描述的 Blazor 服务器框架提供了基础。但我们很少直接使用 SignalR，对于那些需要低延迟消息传递的项目，有更好的替代方案，如 Azure Event Grid 或 Azure Service Bus。

gRPC 是基于 HTTP 的跨平台远程过程调用(RPC)的新兴标准，最初由谷歌(gRPC 中的 g)创建，提供了效率和可伸缩性的优势。gRPC 可能是 Web 服务的未来标准，但它不能在 Web 应用程序中使用，因为它需要对它发送的 HTTP 消息进行低级控制，而这是浏览器不允许的(有一个浏览器库允许 gRPC 通过代理服务器使用，但这破坏了使用 gRPC 的好处)。直到 gRPC 可以在浏览器中使用，它才会包含在 ASP.NET Core 中，但只适用于使用它进行后端服务器之间通信的项目；对于这些项目，存在许多替代协议。当 gRPC 可以在浏览器中使用或成为主要的数据中心协议，本书的未来版本会介绍它。

## 1.2 理解本书

为从本书得到最大的收获，应该熟悉 Web 开发的基础知识，理解 HTML 和 CSS 是如何工作的，并掌握 C#的工作原理。如果没有进行过任何客户端开发(如 JavaScript)，请不要担心。本书的重点是 C#和 ASP.NET Core，随着章节的进展，读者可以获得需要知道的一切。第 5 章总结了 C#

对 ASP.NET Core 开发最重要的特性，如果读者从.NET Core 或者.NET Framework 的早期版本转到 ASP.NET Core，这些特性都很有用。

### 1.2.1　需要什么软件来完成示例?

需要一个代码编辑器(Visual Studio 或 Visual Studio Code)、.NET 核心软件开发工具包和 SQL Server LocalDB。所有这些都可以从微软免费使用，第 2 章包含了所需软件的安装指令。

### 1.2.2　需要什么平台来完成示例?

本书是为 Windows 编写的，使用的是 Windows 10 Pro，但是 Visual Studio、Visual Studio Code 和.NET Core 支持的任何版本的 Windows 都可以工作。其他平台也支持 ASP.NET Core，但本书中的示例依赖于 SQL Server LocalDB 特性，这是 Windows 特有的。如果试图使用另一个平台，可以通过 adam@adam-freeman.com 联系笔者，笔者会给你一些修改示例的一般指示，但有一点要注意，如果你陷入困境，笔者不能提供详细的帮助。

### 1.2.3　源代码下载

读者可访问本书的 GitHub 存储库(https://github.com/apress/pro-asp.net-core-3)下载源代码，也可扫描封底的二维码下载。

### 1.2.4　如果在执行这些示例时遇到问题，怎么办?

首先要做的是回到这一章的开头，重新开始。大多数问题是由于缺少一个步骤或没有完全按照清单执行而引起的。请密切关注代码清单中的重点部分，它会突出显示所需的更改。

接下来，检查该书 GitHub 存储库中包含的勘误表/更正列表。技术书籍是复杂的，尽管笔者和编辑尽了最大努力，错误是不可避免的。检查勘误表列表，获得已知错误列表和解决这些错误的指令。

如果仍然有问题，那么从本书的 GitHub 存储库(https://github.com/apress/pro-asp.net-core-3)下载正在阅读的章节的项目，并将其与项目进行比较。笔者通过遍历每一章来创建 GitHub 存储库的代码，因此项目中的相同文件应该具有相同的内容。

如果示例仍然不能工作，可以联系笔者(adam@adam-freeman.com)寻求帮助。请在电子邮件中说明你正在阅读的是哪本书，是哪一章/例子导致了这个问题。请记住，笔者会收到很多电子邮件，可能不会立即回复。

### 1.2.5　如果发现书中有错误，怎么办?

可以通过向 adam@adam-freeman.com 发电子邮件，向笔者报告错误，不过读者应先检查本书的勘误表/修正列表，本书的 GitHub 库可以在如下地址找到：

```
https://github.com/apress/pro-asp.net-core-3
```

你遇到的错误可能已经在该列表中列出。

### 1.2.6　本书包含的内容

本书试图涵盖大多数 ASP.NET Core 项目都需要的功能。它分为四个部分，每一部分都涵盖

一系列相关的主题。

**第 I 部分：介绍 ASP.NET Core**

本书的这一部分——包括本章——介绍了 ASP.NET Core。除了设置开发环境和创建第一个应用程序之外，还将了解对 ASP.NET Core 开发最重要的 C#特性，和如何使用 ASP.NET Core 开发工具。但是第 I 部分的大部分内容是关于一个名为 SportsStore 的项目的开发，通过这个项目，展示了从开始到部署的实际开发过程，涉及 ASP.NET Core 的所有主要特性，并展示它们是如何结合在一起的。

**第 II 部分：ASP.NET Core 平台**

该部分的章节描述 ASP.NET Core 平台的主要特性，解释如何处理 HTTP 请求，如何创建和使用中间件组件，如何创建路由，如何定义和使用服务，以及如何与 Entity Framework Core 一起工作。解释 ASP.NET Core 的基础知识。理解它们对于有效的 ASP.NET Core 开发是至关重要的。

**第 III 部分：ASP.NET Core 应用程序**

该部分的章节解释如何创建不同类型的应用程序，包括 RESTful Web 服务以及使用控制器和 Razor Pages 的 HTML 应用程序。这些章节还描述易于生成 HTML 的特性，包括视图、视图组件和标签助手。

**第 IV 部分：高级 ASP.NET Core 功能**

本书最后一部分解释如何使用 Blazor 服务器创建应用程序，如何使用实验性的 Blazor WebAssembly，以及如何使用 ASP.NET Core 验证用户身份和授予访问权限。

### 1.2.7 本书未包含的内容

本书没有涵盖基本的 Web 开发主题，如 HTML 和 CSS，也没有详细讲述 C#(第 5 章描述了对 ASP.NET Core 开发有用的 C#特性，使用旧版本.NET 的开发人员可能不熟悉这些特性)。

虽然笔者喜欢在书中钻研细节，但不是每个 ASP.NET Core 特性在主流开发中很有用，笔者必须将本书保持适当的厚度范围内。当笔者决定去掉某个特性时，是因为认为它不重要，或者因为使用所介绍的技术可达到同样的结果。

如前所述，本书没有描述 ASP.NET Core 对 SignalR 和 gRPC 的支持，后续章节提到了没有描述的其他特性，这要么是因为它们不能广泛应用，要么是因为有更好的替代方法可用。每种情况下，笔者都会解释为什么省略了描述，并提供了关于该主题的可供参考的 Microsoft 文档。

### 1.2.8 如何联系作者?

可通过 adam@adam-freeman.com 给笔者发邮件。从第一次在书中公布电子邮件地址以来，到现在已经有好几年了。不能完全肯定这是个好主意，但很高兴笔者这么做了。笔者收到过来自世界各地的电子邮件，读者来自各个行业，大部分邮件都是积极的、礼貌的，收到邮件令人愉悦。

笔者试着迅速回复邮件，但邮件太多，有时还会积压大量邮件，尤其是当笔者埋头写书的时候。笔者总是试图帮助那些被书中的例子困住的读者，尽管读者在联系之前，应按照本章前面描述的步骤来做。

虽然笔者欢迎读者发电子邮件，但对于某些问题，笔者的回答总是否定的。笔者恐怕不会为新公司写代码，帮助完成大学作业，或参与某个开发团队的设计争议。

### 1.2.9　如果你真的喜欢本书?

请发邮件到 adam@adam-freeman.com，让笔者知道。收到快乐读者的来信总是一件让人高兴的事，笔者也很感激读者花时间发送这些邮件。写书很难，这些电子邮件是笔者前进的基本动力。

### 1.2.10　如果本书让人生气，想要抱怨该怎么办?

仍然可以通过 adam@adam-freeman.com 发邮件，笔者仍然会尽力提供帮助。请记住，只有在解释问题是什么、想让笔者怎么做的情况下，笔者才能提供帮助。有时候唯一的结果就是接受本书不是专门为每个人写的，只有放下本书，选择另一本书，事情才会结束。笔者会仔细考虑让人心烦的事情，但在写了 25 年书之后，笔者开始明白，不是每个人都喜欢读我的书。

## 1.3　小结

本章为本书其余部分做了铺垫。简要介绍了 ASP.NET Core，说明了本书的要求和内容，以及如何联系笔者。第 2 章将展示如何准备 ASP.NET Core 开发。