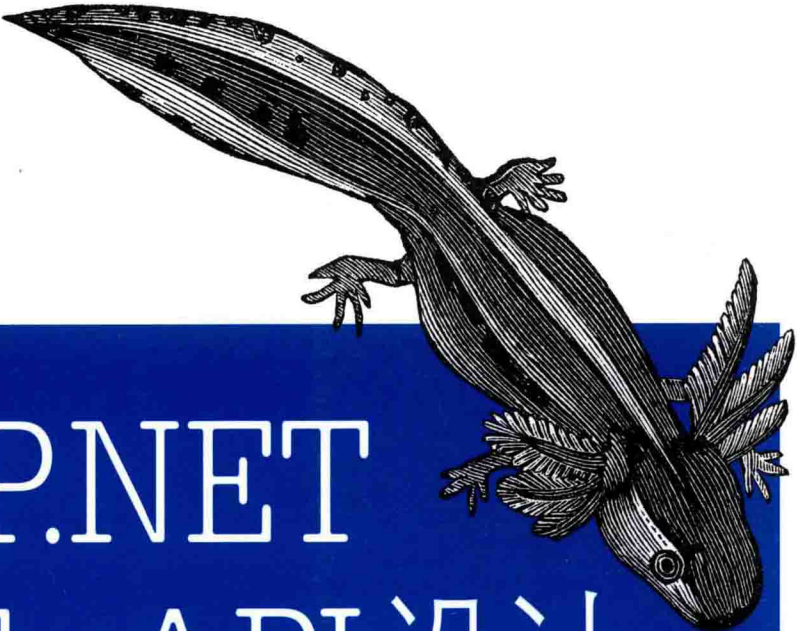


O'REILLY®

TURING

图灵程序设计丛书



ASP.NET Web API设计

Designing Evolvable Web APIs
with ASP.NET

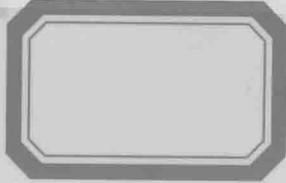


[美] Glenn Block Pablo Cibraro Pedro Félix 著
Howard Dierking Darrel Miller
金迎 译

 人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵程序设计丛书



ASP.NET Web API设计

Designing Evolvable Web APIs with ASP.NET

主要内容

[美] Glenn Block Pablo Cibraro Pedro Félix
Howard Dierking Darrel Miller 著
金迎 译

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

O'Reilly Media, Inc. 授权人民邮电出版社出版

人民邮电出版社

北京

图书在版编目 (CIP) 数据

ASP.NET Web API设计 / (美) 布洛克 (Block, G.)
等著 ; 金迎译. — 北京 : 人民邮电出版社, 2015. 1
(图灵程序设计丛书)
ISBN 978-7-115-37772-2

I. ①A… II. ①布… ②金… III. ①网页制作工具—
程序设计 IV. ①TP393.092

中国版本图书馆CIP数据核字(2014)第277931号

内 容 提 要

本书依托 ASP.NET Web API 阐述 API 设计与开发的通用技术, 是一本全面介绍如何构建真实可演化 API 的实践指南。本书共分为三部分。第一部分介绍 Web/HTTP 和 API 开发的基础知识, 介绍 ASP.NET Web API, 为初学者以及想充分利用 HTTP 的读者建立好的起点。第二部分完整介绍了真实 Web 应用程序的开发, 其内容从设计讲到实现, 全面覆盖客户端与服务端开发。第三部分深入 ASP.NET Web API 的内部机制, 并讲解一些高级的主题 (如安全和可测试性), 加深你的理解, 让读者学会更好地利用 Web API 构建可演化系统。

本书适合使用 .NET、Java、Ruby、PHP、Node 等各平台 API 的开发人员学习参考。

-
- ◆ 著 [美] Glenn Block Pablo Cibraro Pedro Félix
Howard Dierking Darrel Miller
- 译 金 迎
- 责任编辑 李松峰 毛倩倩
- 执行编辑 江 玥
- 责任印制 杨林杰
- ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京鑫正大印刷有限公司印刷
- ◆ 开本: 800×1000 1/16
印张: 28.5
字数: 669千字 2015年1月第1版
印数: 1-3 000册 2015年1月北京第1次印刷
著作权合同登记号 图字: 01-2014-7512号
-

定价: 99.00元

读者服务热线: (010)51095186转600 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京崇工商广字第 0021 号

站在巨人的肩上
Standing on Shoulders of Giants



iTuring.cn

O'Reilly Media, Inc.介绍

O'Reilly Media 通过图书、杂志、在线服务、调查研究和会议等方式传播创新知识。自 1978 年开始，O'Reilly 一直都是前沿发展的见证者和推动者。超级极客们正在开创着未来，而我们关注真正重要的技术趋势——通过放大那些“细微的信号”来刺激社会对新科技的应用。作为技术社区中活跃的参与者，O'Reilly 的发展充满了对创新的倡导、创造和发扬光大。

O'Reilly 为软件开发人员带来革命性的“动物书”；创建第一个商业网站（GNN）；组织了影响深远的开放源代码峰会，以至于开源软件运动以此命名；创立了 Make 杂志，从而成为 DIY 革命的主要先锋；公司一如既往地通过多种形式缔结信息与人的纽带。O'Reilly 的会议和峰会集聚了众多超级极客和高瞻远瞩的商业领袖，共同描绘出开创新产业的革命性思想。作为技术人士获取信息的选择，O'Reilly 现在还将先锋专家的知识传递给普通的计算机用户。无论是通过书籍出版，在线服务或者面授课程，每一项 O'Reilly 的产品都反映了公司不可动摇的理念——信息是激发创新的力量。

业界评论

“O'Reilly Radar 博客有口皆碑。”

——*Wired*

“O'Reilly 凭借一系列（真希望当初我也想到了）非凡想法建立了数百万美元的业务。”

——*Business 2.0*

“O'Reilly Conference 是聚集关键思想领袖的绝对典范。”

——*CRN*

“一本 O'Reilly 的书就代表一个有用、有前途、需要学习的主题。”

——*Irish Times*

“Tim 是位特立独行的商人，他不光放眼于最长远、最广阔的视野并且切实地按照 Yogi Berra 的建议去做了：‘如果你在路上遇到岔路口，走小路（岔路）。’回顾过去 Tim 似乎每一次都选择了小路，而且几次都是一闪即逝的机会，尽管大路也不错。”

——*Linux Journal*

Glenn Block 曾在 ASP.NET 团队工作，负责开发 ASP.NET Web API 的早期版本，现在就职于 Splunk，改善开发人员访问大数据的能力。作为一名拥有近 20 年经验的职业程序员，Block 一直在努力改进开发人员的工作。Block 醉心编码，据说“极少睡觉”，还是云开发的积极支持者，在微软 Windows Azure 对 OSS 栈的支持中起了关键作用。Block 是 Node.js 和 .NET OSS 项目的热心参与者、社区支持者，还经常在国际会议上发言。Block 与妻女现居住在西雅图。

Pablo Cibraro 是 Tellago 公司的一位软件架构师，也是一位享有国际声誉的专家，拥有十余年系统架构和使用微软技术实施大型分布式系统的经验。Cibraro 在过去几年间与微软公司的模式和实践团队直接合作，为使用 Web 服务、Web 服务增强（WSE）和 WCF 构建面向服务的应用程序提供示例程序、模式和指南。

Pedro Félix 居住在葡萄牙里斯本，教授和研究编程与计算机安全相关的课题。

Howard Dierking 是微软 WCF Web API 团队的产品经理，主要负责 AppFabric。在此之前，Dierking 在微软担任过其他多个角色，其中有《MSDN 杂志》的主编和 Microsoft Learning 的产品规划专员。

Darrel Miller 是 Tavis 软件公司的合伙人，这是一家专为制造业提供软件解决方案和服务的公司。Miller 的工作是向人们展示如何在业务应用程序中采用 REST 架构风格。

关于封面图

本书封面上的动物是疣螈（学名 *Triturus cristatus*），又名北螈或巨冠蝾螈。这种两栖动物分布于欧洲北部，从英国一直到黑海。疣螈是生活在不列颠群岛上的三种蝾螈中最大最稀有的一种，在当地受到生物多样性行动计划的保护。生物多样性行动计划致力于将濒危物种进行统计并形成保护计划。

疣螈大部分时间呆在陆地上，但是会回到池塘中进行繁殖。幼虫大约三周后孵化，并在水下生长一段时间，在四个月大时经历蜕变，成为呼吸空气的青年疣螈，离开池塘上岸居住。在陆地上，疣螈捕食昆虫及其幼虫。成年疣螈还会在池塘中捕食其他蝾螈、蝌蚪、幼年青蛙、昆虫或田螺。

因其防御能力相对较弱，疣螈喜欢居住在有植被覆盖的陆地上，例如灌木丛、草丛和茂密的丛林。雌性疣螈体积比雄性大，长度可以达到 15 厘米。雄性和雌性疣螈都具有相同的颜色图案：背部和侧面为深灰到黑色，腹部为黄色或橘红色，并有黑色斑点。在繁殖期，雄性疣螈背部生出锯齿状颈脊，与雌性疣螈外观不同。

这些疣螈从 10 月到 3 月冬眠，沉睡于繁殖地池塘底部淤泥中的木头和石头下面。通常，疣螈每年都返回同一个繁殖地，一般不会离开出生地半英里的范围。虽然有些疣螈可以活 30 年之久，但是在野外大部分只能存活约 10 年。

序

1989年3月，当Tim Berners-Lee在欧洲核子研究组织（CERN，<http://info.cern.ch/>）首次提出Web的概念（<http://www.w3.org/History/1989/proposal.html>）时，就引发了一场创造性和机遇的社交革命。这场革命从此横扫全球，改变了社会的运作方式和人们的交互方式，也改变了我们如何看待自己作为个体在社会中承担的角色。

但是，Berners-Lee还引发了一场影响同样深远的技术革命，改变了工程师们为Web设计思考和构建软硬件系统的方式。Web服务器的概念从单个计算机变成了全球云端基础设施中一个完全虚拟的部分。在云端，我们可以在任何需要的地方进行计算。同样，Web客户端也从传统的安装了浏览器的桌面个人电脑，变成了无数可以感知物理世界并与之交互的设备，这些设备通过云端的Web服务器与其他设备相连。

如果我们想想Web经历的变化，激动人心的不仅仅是这些变化发生的令人目眩的速度，还在于这些变化的发生没有任何集中的控制或协调。也就是说，Web经历的是演化。为了适应新的需求，新的想法和解决方案不断涌现，与旧想法进行竞争。有时候，新想法胜出，存活下来；有时候，新想法落败，自行消亡。

演化天生就是Web不可或缺的一部分。和大自然的演化一样，如果Web中的某些组件适应变化的能力较强，就更有机会保持活力，不断发展。

不仅Web服务器和客户端发生了变化，服务器和客户端之间的交互方式也在发生巨大的改变。过去，Web服务器提供HTML，由客户端展示为Web页面；类似地，Web客户端将HTML表单提交给服务器处理，无论是处理比萨饼订单，插入一篇博客文章，还是更新缺陷管理系统中的一个问题。

这种交互模型主要使用GET和POST方法，实际上只用到了HTTP的一部分功能。但是，HTTP从一开始就定义了一个范围更广的应用程序模型，全面支持与数据的交互和对数据的操控。例如：除了经典的GET和POST方法，HTTP还定义了PUT、DELETE和PATCH方法，可

以编程操控资源，并与资源进行交互。

这就是我们用到 Web API 的地方：利用 Web API，Web 服务器可以提供完整的 HTTP 应用程序模型，支持对资源的编程访问，使客户端能够在多种不同情况下，以一致的方式与数据进行交互，并对其进行操控。

推动人们转向 Web API 的关键因素有两个：HTML5 和移动应用。HTML5 和移动应用都利用客户端平台的计算能力，提供引人入胜的流畅体验，同时通过后台 Web API 获取和操控数据。概括地说，过去 Web 服务器只提供静态 HTML，而现在也提供 Web API，因此客户端可以使用 HTTP 应用程序模型的全部功能，进行程序化交互。这本书要解决的问题，就是如何构建出这样的 Web API。简言之，任何人要构建针对 HTML5 应用程序以及移动应用的 Web API，都应该读一读本书。本书不仅很好地介绍了 Web API，还提供很多实用指南，指导大家使用 ASP.NET Web API 构建 Web API。此外，本书还详尽地介绍了 ASP.NET Web API 的工作原理。你也可以以本书为参考，了解如何通过 HTTP 消息处理程序、格式化程序等，对 ASP.NET Web API 进行扩展。

本书不仅包括代码展示和框架说明，还介绍了一些强大的技术，例如：TDD（Test-Driven Development，测试驱动开发）和 BDD（Behavior-Driven Development，行为驱动开发），帮助你编写应用程序，并测试、验证程序功能是否符合预期。

更棒的是，本书不仅提供了关于如何构建 Web API 的“当前”指南，还引导你逐步了解如何设计一个随着需求和约束变化演化的 Web API。解决可演化性问题的想法渗透到了 Web 运作方式的核心。

在这种环境下构建一个有效运行的 Web API 并非易事。但有一点很明确，你必须从一开始就接受一个观点：任何 Web API 都必将发生改变，而且没有人能在某一时刻控制环境中的所有因素。也就是说，如果你设计了系统的一个新版本，就把旧版本抛弃，那么必定会失去已有用户或者导致问题——你必须逐步对系统进行改进，既要继续支持旧客户端，同时又要向新客户端提供新功能。

但是，构建一个灵活的可演化软件仍然是个难题。本书极佳地阐述了如何构建可以随着需求变更和演化的现代 Web 应用程序，其介绍方式是将 Web API 与超媒体结合，这是 Web 应用程序的一个激动人心的新方向。

超媒体既是一个新概念，也是一个旧概念。我们都惯于浏览 Web 页面，寻找信息，然后点击一个链接，打开新的页面，获得更多的信息和链接，深入了解感兴趣的方面。随着信息发生改变或演化，Web 页面可以加入新链接，或者修改已有链接以反映这些变化。这些新链接可以为你提供新信息，深入更多的领域。

将 Web API 与超媒体相结合，你将得到一个强大的模型，应用程序可以像 Web 页面一样变

化，适应和调整程序与服务器交互的方式。现在，客户端不再有固定的操作流程，而是根据可用的链接修改自己的行为，以进行演化；简言之，客户端能够适应变化。

对此，本书提供了一个全面的概览，介绍设计可适应双方（提供商和消费者）需求变化的 Web API 的最先进方法。通过介绍相关概念，例如使用测试驱动开发构建超媒体驱动的 Web API，本书为需要构建 Web API 的开发者提供了一个很好的起点。

作为 ASP.NET Web API 开发团队的一员，我很高兴与本书的作者一起工作。本书的作者非常优秀，他们不但都具有构建框架的经验，而且拥有构建基于 HTTP 概念的实际系统的丰富实战经验。本书的作者提出了很多极有价值的意见和建议，帮助 ASP.NET Web API 成为构建现代 Web 应用程序的流行框架。

我与 Glenn Block 的合作特别愉快，Block 很早就加入了 ASP.NET Web API 项目，极大推动了项目对社区参与的重视，以及对依赖注入、测试驱动开发和超媒体重要性的认识。没有 Block 的贡献，ASP.NET Web API 就不会达到今天的高度。

如果你正在或者考虑构建 Web API，那么本书不仅可以作为一个学习工具，而且可以用作实际的指南，帮助你基于 ASP.NET Web API 构建现代 Web 应用程序。本书提供了大量的信息和指导原则，将教你以一种新颖的方式看待复杂问题，在设计中时刻考虑可演化性。我自己非常期待看到本书在未来如何演化！

——Henrick Frystyk Nielsen

前言

为什么要阅读本书

Web API 开发呈现爆炸式增长。各家公司都在投资构建可以通过 Web 使用多种客户端访问的系统。想想你经常光顾的网站，这些网站很可能已经提供了访问 API。创建一个使用 HTTP 进行通信的 API 非常简单，而挑战在第一个版本部署后才会出现。实际上，HTTP 协议的制定者对这个问题，以及如何设计可演化 API 已经进行了周详的考虑。媒体类型和超媒体就是设计可演化 API 的核心。但是，很多 API 开发者并未对此加以考虑或利用，部署的 API 没有合理地使用 HTTP 协议，客户端严重依赖 API 的具体实现。这样的 API 很难进行演化，极易破坏客户端功能。为什么会出现这样的情况呢？因为人们经常觉得，从工程角度看，这是实现功能的最简单、最直接的做法。但是，从长期看，这种做法违反直觉，与 Web 自身的基本设计原则背道而驰。

本书的目标读者是希望设计出适应长期变化的 API 的开发者。变化是不可避免的：你今天构建的 API 将会演化。因此，问题不是“要不要”，而是“如何”设计可演化的 API。你在项目早期做出的决定（或者未做出的决定）会极大地影响以下问题的答案。

- 添加一个新功能，会破坏现有的客户端，强制现有客户端升级并重新部署吗？或者，现有客户端还能继续工作？
- 如何保障 API 的安全？能够使用较新的安全协议吗？
- API 可以扩展规模，满足用户需求吗？或者必须重新进行架构设计？
- 能够支持未来出现的新客户端和新设备吗？

你在设计时可以考虑这些问题。初看起来，这似乎是预先做大量设计（Big Design Up Front）或者瀑布方法。其实不然。你不需要在构建系统前做出完备的设计，也不需要进行分析。有些决定的确需要预先做出，但这些决定处于较高的层次，关系到整体设计。要做出这些决定，你并不理解或预知系统的每个方面。实际上，这些决定奠定了迭代演化的

基础，在随后构建系统时，你可以在此基础上，采取各种方式不断强化自己的目标。

本书更偏重应用而非理论。我们希望你读完本书之后，获得构建真实可演化系统的能力。为了达到这个目的，开篇将介绍 Web 和 Web API 开发的一些必知内容，然后从设计到实现，逐步介绍如何使用 ASP.NET Web API 创建一个新 API。这个 API 的实现将覆盖一些重要的主题，例如：如何使用 ASP.NET Web API 实现超媒体、如何执行内容协商。我们将演示这个 API 在部署后如何实际进行演化，还将讨论如何使用既有实践（例如：验收测试、测试驱动开发）和技术（例如：控制反转）提高代码的可维护性。最后，我们将深入 Web API 的内部，帮助你加深理解，更好地利用 Web API 构建可演化系统。

预备知识

要充分理解本书的内容，你应该是一位开发人员，拥有使用 .NET 3.5 或更高版本开发 C# 应用程序的经验。如果你还具有构建 Web API 的经验，那就更好了。在开发 API 时使用过哪种框架并不重要，重要的是应该熟悉相关的概念。阅读本书并不需要 ASP.NET Web API 或 ASP.NET 经验，但熟悉 ASP.NET MVC 的确会对你很有帮助。

如果你不是一位 .NET 开发人员，那么本书也有适合阅读的内容。我们编写本书时设定了一个具体的目标，要使书中大部分内容关注 API 设计和开发的通用技术，不与 ASP.NET Web API 直接相关。因此，我们认为，无论你的开发背景是什么（Java、Ruby、PHP、Node 等），都可以通过本书前两部分的大部分内容学习 API 的开发。

漫游指南

在开始你的阅读旅程之前，这里有一些漫游本书内容的指南。

- 第一部分主要对 Web API 开发进行介绍。这部分覆盖了 Web/HTTP 和 API 开发的基础知识，介绍 ASP.NET Web API。如果你刚刚接触 Web API 开发 /ASP.NET Web API，那么这部分是一个很好的开始。如果你已经在使用 ASP.NET Web API（或其他 Web API 框架），但是想更多地了解如何充分利用 HTTP，这部分也是一个很好的起点。
- 第二部分关注真实世界的 Web API 开发。这部分完整介绍了一个真实世界中 Web 应用程序的开发，从设计到实现，覆盖客户端和服务端。如果你已经对 Web API 开发颇为熟悉，希望很快开始构建应用程序，那么可以直接从第二部分开始阅读。
- 第三部分是一个相当全面的参考资料，详细介绍了 ASP.NET Web API 各部分的内部机制。这部分还覆盖了一些较为高级的主题，例如：安全和可测试性。如果你已经在使用 ASP.NET Web API 构建应用程序，希望了解如何将 Web API 自身的功能发挥到极致，请从第三部分开始阅读。

本书内容

本书分为三部分。

第一部分 基础知识

- 第 1 章 因特网、万维网和 HTTP 协议

这一章开篇简单回顾了万维网和 HTTP 协议的历史，然后概要介绍了 HTTP 协议。你可以把这一章看成 HTTP “傻瓜指南”。这一章提供了你需要了解的关键知识，以免去你阅读整个 HTTP 规范的辛苦。

- 第 2 章 Web API

这一章首先大致介绍了 Web API 开发的历史背景，然后讨论 API 开发的精髓，从核心概念开始，一直深入探讨到设计 API 的风格和方法。

- 第 3 章 ASP.NET Web API 基础

这一章讨论了 ASP.NET Web API 框架背后的基本驱动因素，然后介绍 ASP.NET Web API 的基础知识，以及 .NET HTTP 编程模型和客户端。

- 第 4 章 处理架构

这一章将简要介绍一个 HTTP 请求在 ASP.NET Web API 中进行处理的生命周期。你将了解到处理 HTTP 请求和响应的不同方面涉及的各个不同部分。

第二部分 真实世界的 API 开发

- 第 5 章 应用程序 + 第 6 章 媒体类型选择与设计

这两章讨论了问题跟踪应用程序的整体设计，涵盖了设计相关的几个重要主题，其中有：媒体类型选择和设计，以及超媒体。

- 第 7 章 构建 API + 第 8 章 改进 API

这两章展示了如何使用 ASP.NET Web API 实现和改进超媒体驱动的问题跟踪 API，并介绍了如何使用行为驱动的开发方式。

- 第 9 章 构建客户端

这一章介绍如何构建问题跟踪 API 的一个超媒体客户端。

第三部分 Web API细节

- 第 10 章 HTTP 编程模型

这一章将深入讨论 ASP.NET Web API 依托的 .NET HTTP 编程新模型。

- 第 11 章 托管

这一章介绍了 ASP.NET Web API 适用的所有托管模型，其中有：自托管、IIS 以及新的 OWIN 模型。

- 第 12 章 控制器和路由

这一章深入探讨了 Web API 路由的工作机制，以及控制器如何运作。

- 第 13 章 格式化程序和模型绑定 / 第 14 章 HttpClient

这两章介绍了关于模型绑定和使用新的 HTTP 客户端类的所有需知内容。

- 第 15 章 安全 / 第 16 章 OAuth 2.0 授权框架

这两章介绍了 ASP.NET Web API 的整体安全模型，然后详细讨论了如何在 API 中实现 OAuth 规范。

- 第 17 章 可测试性

这一章将介绍如何以测试驱动的方式，在 ASP.NET Web API 中进行开发。

排版约定

本书使用了下述排版约定。

- 楷体

标示新术语。

- 等宽字体 (*constant width*)

表示程序片段，也用于在正文中表示程序中使用的变量、函数名、命令行代码、环境变量、语句和关键字等元素。

- 等宽粗体 (***constant width bold***)

表示应该由用户逐字输入的命令或者其他文本。

- 等宽斜体 (*constant width italic*)

表示应该由用户输入的值或根据上下文决定的值替换的文本。



这个图标代表提示或建议。



这个图标代表重要说明。



这个图标代表警告或提醒。

使用代码示例

读者可以在这里下载本书随附的资料（代码示例、练习题等）：<https://github.com/webapibook>。讨论本书的论坛地址为：<https://groups.google.com/forum/#!forum/webapibook>。

希望本书能够助你一臂之力。也许你需要在自己的程序或文档中用到本书中的代码。除非大段大段地使用，否则你不必与我们联系取得授权。例如，无需请求许可，你就可以用本书中的几段代码写成一个程序。但是销售或者发布 O'Reilly 图书中代码的光盘则必须事先获得授权。引用书中的代码来回答问题也无需获得授权。将大段的示例代码整合到你自己的产品文档中则必须经过许可。

使用我们的代码时，希望你能标明它的出处，但不强求。出处一般包括书名、作者、出版商和 ISBN，例如：*Designing Evolvable Web APIs with ASP.NET* by Glenn Block, Pablo Cibraro, Pedro Felix, Howard Dierking, Darrel Miller (O'Reilly). Copyright 2012 Glenn Block, Pablo Cibraro, Pedro Felix, Howard Dierking, and Darrel Miller, 978-1-449-33771-1。

如果还有关于使用代码的未尽事宜，你可以随时与我们联系：permissions@oreilly.com。

Safari® Books Online



Safari Books Online (<http://www.safaribooksonline.com>) 是应需而变的数字图书馆。它同时以图书和视频的形式出版世界顶级技术和商务作家的专业作品 (<http://www.safaribooksonline.com/content>)。

Safari Books Online 是技术专家、软件开发人员、Web 设计师、商务人士和创意人士开展调研、解决问题、学习和认证培训的第一手资料。

对于组织团体 (<http://www.safaribooksonline.com/organizations-teams>)、政府机构 (<http://www.safaribooksonline.com/government>) 和个人 (<http://www.safaribooksonline.com/individuals>), Safari Books Online 提供各种产品组合 (<http://www.safaribooksonline.com/subscriptions>) 和灵活的定价策略。用户可通过一个功能完备的数据库检索系统访问 O'Reilly Media、Prentice Hall Professional、Addison-Wesley Professional、Microsoft Press、Sams、Que、Peachpit Press、Focal Press、Cisco Press、John Wiley & Sons、Syngress、Morgan Kaufmann、IBM Redbooks、Packt、Adobe Press、FT Press、Apress、Manning、New Riders、McGraw-Hill、Jones & Bartlett、Course Technology 以及其他几十家出版社 (<http://www.safaribooksonline.com/publishers>) 的上千种图书、培训视频和正式出版之前的书稿。要了解 Safari Books Online 的更多信息, 我们网上见 (<http://www.safaribooksonline.com/>)。

联系我们

请把对本书的意见和疑问发送给出版社。

美国:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472

中国:

北京市西城区西直门南大街 2 号成铭大厦 C 座 807 室 (100035)
奥莱利技术咨询(北京)有限公司

O'Reilly 的每一本书都有专属网页, 你可以在那儿找到本书的相关信息, 包括勘误表、示例代码以及其他信息。本书的网站地址是:

<http://oreil.ly/designing-api>

如果你对本书有一些建议或技术上的疑问, 请发送电子邮件至 bookquestions@oreilly.com。

要了解更多 O'Reilly 图书、培训课程、会议和新闻的信息, 请访问以下网站:

<http://www.oreilly.com>

我们在 Facebook 的地址如下: <http://facebook.com/oreilly>

请关注我们的 Twitter 动态: <http://twitter.com/oreillymedia>

我们的 YouTube 视频地址如下: <http://www.youtube.com/oreillymedia>

致谢

完成这本书, 实际的工作量远超我们的预计。首先要感谢我们的妻子和孩子, 他们给予了