TURING 图灵程序设计丛书

MANNING

OAuth 2 IN ACTION

OAuth 2 实战

[美] 贾斯廷·里彻
[瑞士] 安东尼奥·桑索 ◎著

杨 鹏 ◎译

- 详尽介绍OAuth生态系统及其工作原理
- OAuth工作组重要成员执笔
- 掌握Web安全知识必读

中国工信出版集团

人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵程序设计丛书

OAuth 2 IN ACTION

OAuth 2 实战

[美] 贾斯廷 · 里彻
[瑞士] 安东尼奥 · 桑索 ◎著

杨 鹏 ◎译

人民邮电出版社
北 京

图书在版编目（CIP）数据

OAuth 2实战 / (美) 贾斯廷·里彻 (Justin Richer)，(瑞士) 安东尼奥·桑索 (Antonio Sanso) 著；杨鹏译. -- 北京：人民邮电出版社，2019.4
（图灵程序设计丛书）
ISBN 978-7-115-50937-6

Ⅰ. ①O… Ⅱ. ①贾… ②安… ③杨… Ⅲ. ①计算机网络－安全技术－研究 Ⅳ. ①TP393.08

中国版本图书馆CIP数据核字(2019)第041374号

内 容 提 要

本书深入探讨OAuth的运行机制，详细介绍如何在不安全的网络环境下正确使用、部署OAuth，确保安全认证，是目前关于OAuth最全面深入的参考资料。书中内容分为四大部分，分别概述OAuth 2.0协议，如何构建一个完整的OAuth 2.0生态系统，OAuth 2.0生态系统中各个部分可能出现的漏洞及其如何规避，以及更外围生态系统中的标准和规范。

本书适合所有想了解OAuth工作原理及其应用的读者。

◆ 著　[美] 贾斯廷·里彻 [瑞士] 安东尼奥·桑索
译　杨　鹏
责任编辑　谢婷婷
责任印制　周昇亮

◆ 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：18.5
字数：448千字　2019年4月第1版
印数：1－3 000册　2019年4月北京第1次印刷
著作权合同登记号　图字：01-2017-8625号

定价：89.00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：京东工商广登字20170147号

站在巨人的肩上

Standing on Shoulders of Giants

iTuring.cn

站在巨人的肩上

Standing on Shoulders of Giants

iTuring.cn

版权声明

序

没有什么比一片空白更让人畏惧了，它盯着你，你却一筹莫展。

你并不是不知道要做什么，相反，你对自己的构想有清晰的认识。你甚至可以想象到，当老板和客户见到你的杰作时脸上露出的满意笑容。但问题是，你的面前是一片空白。

所以，你伸手去拿工具。鉴于你正在阅读这本书，你很可能是一名开发人员或者身份认证方面的专业人士。不管怎样，你都知道安全是至关重要的，并希望自己的杰作得到安全保护。

说到 OAuth，你应该知道，它可以用于保护资源，尤其是 API。它的应用非常广泛，而且看起来无所不能。问题恰恰在于它的无所不能使其很难驾驭。这又是一片空白。

再说说这本书和两位合著者。当你手中有一件无所不能的工具却无从下手时，最好的办法就是将它利用起来。这本书不仅解释了 OAuth 的用途，还会引导你完成整个应用流程。最终，你不仅会对 OAuth 这一工具有非常深入的了解，而且面前将不再是一片空白——你做好了实现头脑中伟大构想的准备。

OAuth 是一个非常强大的工具，它的强大来自其灵活性，灵活性通常意味着它不仅能够完成你的构想，而且也会带来安全问题。OAuth 管理 API 的访问权限，守护着重要数据，所以最关键的是避免反模式，运用最佳实践，以安全的方式使用它。换句话说，虽然它的灵活性让你可以以任何方式使用和部署它，但并不意味着你应该那样随意。

关于 OAuth，还有一件要提到的事情——你不是为了用 OAuth 而去用它。你是想用它来做一些别的事情——很可能是要将一组 API 调用精心组合起来，以实现一个绝妙的点子。你或许正在头脑中勾勒一幅完整图景，正思考如何实现自己的杰作。OAuth 可以帮助你以更安全的方式实现它。

幸运的是，两位合著者提供了一份实用指南，告诉我们什么该做，什么不该做。他们让你一箭双雕，同时实现“我想搞定这件事情”和“我想确认这样做是安全的”两种想法。

随着空白被填补，你最终会实现自己的杰作并交到客户手中，同时意识到这项工作其实并不难。

Ian Glazer

Salesforce 公司身份管理高级总监

前　言

我是 Justin Richer，日常主要从事顾问工作，虽然假装自己是个安全怪咖，但我并不是科班出身。我的专业背景是协作技术，主要研究如何让人们借助计算机协同工作。即便如此，我也早就开始使用 OAuth 了。我用先前的 OAuth 1.0 实现了好几个服务器和客户端，将它们用在我当时开发的协作系统中。正是从那个时候我开始体会到，如果应用架构想要在真实环境中立于不败之地，那么一个优良的、易实施的、易用的安全系统便不可或缺。大约也是在那个时候，我参加了早期的 Internet Identity Workshop 会议。在会议上大家讨论了下一代 OAuth，决定将 OAuth 1.0 在实际应用中的经验教训作为下一代 OAuth 的构建基础。在互联网工程任务组（IETF）开始开发 OAuth 2.0 时，我加入了这个小组并首次深度参与了讨论。几年之后，我们将规范制定了出来。虽然称不上完美，但它表现得很不错，人们对它青睐有加。

我一直留在 OAuth 工作组中，并担任了 Dynamic Registration（RFC 7591 和 RFC 7592）和 Token Introspection（RFC 7662）这两个 OAuth 扩展规范的编辑。如今，我是 OAuth Proof of Possession（PoP）这一套规范的编辑以及其中某些部分的作者，还担任了多个 OAuth 配置规范（profile）、扩展以及相关协议的技术编辑。我还参与制定了 OpenID Connect 核心规范，并和我的团队一起实现了广受好评的 MITREid Connect，它是基于 OAuth 和 OpenID Connect 的服务端与客户端套件。我突然发现，自己在向许多不同的受众谈论 OAuth 2.0，并在各种各样的系统上实现它。我教过课，做过演讲，还写过一些关于 OAuth 2.0 的文章。

所以，当 Antonio Sanso—— 一位受人尊敬的安全研究者——邀请我合著本书的时候，我们一拍即合。在搜寻了市面上有哪些关于 OAuth 2.0 的书之后，我们发现一本喜欢的都没有。我们找到的大部分书都是针对具体服务的，例如，如何写一个能和 Facebook 或 Google 交互的 OAuth 客户端，或者如何使用 GitHub API 对本地应用授权。如果你所关心的只是这方面的内容，那资料真是相当丰富。但我们没有看到有哪本书向读者介绍整个 OAuth 系统，解释其工作原理，指出其弱点、局限性以及优势。我们决定要以全方位的视角来写这样一本书，并且力求完美。因此，本书不会涉及任何与现实中特定 OAuth 服务商的交互，也不会详细讨论特定的 API 或行业领域。相反，本书的焦点在于 OAuth 本身的原理，希望让读者看到：当曲柄转动时，每一个齿轮是如何啮合的。

我们构建了一个代码框架，希望读者将注意力集中在 OAuth 的核心部分，而不要过度地陷入平台实现的细节。毕竟，这不是一本教读者“如何在最新的平台上实现 OAuth”的书，而是想以本书“讲解 OAuth 2.0 的工作原理并让读者能在任何平台上使用它”。所以，我们基于 Express.js 构建了一个相对简易的 Node.js 框架。为了尽可能地消除平台特异性，我们大量使用了库。尽管如此，JavaScript 还是 JavaScript，有些特异之处仍然会不时出现，任何平台都会这样。但我们希

望，读者能将本书所用的方法和理论应用到自己所喜爱的语言、平台和架构中去。

回顾历史，我们是如何走到现在的？故事开始于 2006 年，当时包括 Twitter 和 Ma.Gnolia 在内的很多 Web 服务公司，都有很多应用是互通的，他们希望能让用户将这些应用统一地连接起来。当时，此类连接都是通过在远程服务器上向用户索要凭据并将凭据传送到 API 上来实现的。然而，为方便登录，这些网站都使用了一项分布式身份认证技术——OpenID。这就导致无法将用户名和密码用于 API。

为了解决这个问题，开发人员试图发明一个协议，允许用户将 API 访问授权出去。新的协议基于多个具有同样思想的专有实现，包括 Google 的 AuthSub 以及 Yahoo!的 BBAuth。在这些实现中，客户端应用只要获得用户的授权并得到一个令牌，就能使用这个令牌访问远程 API。这些令牌的发放都包含公共和私有的部分，并且该协议使用了一种新型的（现在看来是脆弱的）加密签名机制，使得它可以在非 TLS HTTP 连接上使用。他们称这个协议为 OAuth 1.0，并将其作为一项 Web 开放标准发布。它很快就获得响应，出现了多种语言对这一标准的自由实现。这一标准表现优秀，深受开发人员喜爱，甚至一些大型互联网公司也很快弃用了自己的专有机制，起初正是这些专有机制启发了 OAuth。

和许多新的安全协议一样，OAuth 1.0 在早期就被发现有一个缺陷。为了修复这个会话固化漏洞，OAuth 1.0a 诞生了。这个版本后来作为 RFC 5849 被编入了 IETF。在那个时候，OAuth 协议的社区开始成长，新的用例被开发和实现。其中一些用例将 OAuth 用在了其原本并未有意适用的场景，但这些对 OAuth 的非常规使用也比现有的其他可用方案表现得更好。然而，OAuth 1.0 只是一个单体协议，意图以不变应万变，用同一种机制来应对所有场景，在有些场景下使用它是有风险的。

RFC 5849 发布后不久，Web Resource Access Protocol（WRAP）就发布了。这份协议采用了 OAuth 1.0a 的核心——客户端、委托、令牌——并对它们进行了扩展，以适应不同的需求。WRAP 废除了 OAuth 1.0 中很多令人困惑和容易出问题的部分，比如自定义签名算法机制。经过社区的多次讨论，WRAP 被确定为新的 OAuth 2.0 协议的基础。OAuth 1.0 是单体的，而 OAuth 2.0 是模块化的。OAuth 2.0 的模块化使其成为了一个框架，能够部署和运用在 OAuth 1.0 已经实践过的所有场景中，但并不会让协议的核心内容发生扭曲。本质上来说，OAuth 2.0 在基础层面提供了设计规范。

2012 年，IETF 批准了 OAuth 2.0 核心规范，但是留给社区的工作还远未完成。规范被模块化地分成互补的两个部分：RFC 6749 详细说明了如何获取令牌，RFC 6750 则详细说明了如何在受保护的资源上使用一种特定类型的令牌（bearer 令牌）。另外，RFC 6749 的核心部分还详细阐明了多种获取令牌的方式，并提供了一种扩展机制。OAuth 2.0 定义了 4 种许可类型，分别适用于不同的应用类型，而不是单单定义一种复杂的方法来适应不同的部署模型。

如今，OAuth 2.0 已经是互联网上首选的授权协议。它被广泛使用，从大型互联网公司到小型创业公司，几乎所有的地方都在使用它。由基于 OAuth 2.0 构建的扩展、配置规范和完整协议组成的生态系统已经出现，人们也不断探索出对这一基础技术更多新颖、有趣的应用方式。本书的目标是帮助读者不仅理解 OAuth 2.0 是什么以及它的工作原理，而且还要知道如何以最佳的方式来用它解决问题、构建系统。

Justin Richer

致　谢

创作本书的过程真的犹如一段充实的旅程。当我们启动写书计划并列出大纲时，就意识到所需付出的精力将远超预期。事实证明当时的判断是正确的，也非常高兴我们终于可以写致谢语了，感谢所有帮助我们完成本书的人。我们可能无法在此列出所有人的名字，所以即使您的名字没有出现在这里，也请接受我们诚挚的谢意。

首先要感谢 IETF 的 OAuth 工作组以及外围的 OAuth 及开放标准社区，没有他们的付出和鼓励，本书不可能完成。特别是 John Bradley 和 Hannes Tschofenig，他们为本书提出了许多宝贵的意见。感谢社区中的 Ian Glazer、William Dennis、Brian Campbell、Dick Hardt、Eve Maler、Mike Jones 和其他许多人给予我们鼓励，并在网上提供了很多重要信息。感谢 Aaron Parecki 为我们提供了 oauth.net 上的空间，让我们不仅可以讨论本书，还发布了专题文章，其中包括第 13 章的早期版本。特别要感谢 Ian 为本书作序，并认可我们的工作。

如果没有 Manning 出版团队的帮助和付出，本书就不可能面世。我们有出色的编辑团队和支持人员，他们是 Michael Stephens、Erin Twohey、Nicole Butterfield、Candace Gillhoolley、Karen Miller、Rebecca Rinehart、Ana Romac，特别是编辑 Jennifer Stout 非常了不起。感谢 Ivan Kirkpatrick、Dennis Sellinger 和 David Fombella Pombal，他们负责校对技术部分。非常感谢所有在本书还处于 MEAP 状态就预订的人，你们的早期反馈至关重要，让我们得以尽可能地以最高水准呈现本书。

还要感谢同行审稿人，他们审阅了本书各个阶段的手稿，并提出了宝贵意见，他们是：Alessandro Campeis、Darko Bozhinovski、Gianluigi Spagnuolo、Gregor Zurowski、John Guthrie、Jorge Bo、Richard Meinsen、Thomas O'Rourke 以及 Travis Nelson。

Justin Richer

我最最应该感谢的是我的合著者 Antonio Sanso。他的安全和加密专业知识远超我的想象，我非常荣幸能与他合作。写作本书最开始是他的提议，我们一起协作，最终得以完成。

感谢我的朋友 Mark Sherman 和 Dave Shepherd，他们在我开始写书之前就已成功出版过技术书。他们的出版经验给了我很大帮助，就像是隧道尽头的一盏明灯。感谢 John Brooks、Tristan Lewis 和 Steve Moore，他们能让我蹦出一些想法和词句，尽管他们当时并没有觉察到。

非常感谢我的客户容忍我在过去的一年中因投入写作而不时地玩消失。特别感谢 Debbie Bucci 和 Paul Grassi，他们出色的工作计划让我获得了使本书落地的第一手经验。

对同事和朋友 Sarah Squire 的感谢，我无以言表。是她最初向我推荐了本书练习中使用的 Node.js 框架，而且是她去办公用品商店打印了本书的第一个印刷版。总之，她对这个项目的鼓励、支持、批评和热情是无与伦比的，如果没有她，本书不会完成。

最后，也是最重要的，我要诚挚地感谢我的家人。我的妻子 Debbie，以及孩子 Lucien、Genevieve 和 Xavier 给予我无尽的耐心。无数个深夜和周末，我将自己关在办公室里，与世隔绝，我敢肯定他们都已开始怀疑我是否还出得来。但现在，我很高兴地说，我们有大把的时间来玩乐高了。

Antonio Sanso

编写本书的过程是一段美妙的旅途，当写到这一部分时，我心怀愉悦和满足。与所有旅途一样，重要的不是目的地，而是沿途的风景。没有周围的人对我的帮助，我不可能参与完成本书。

感谢我的雇主 Adobe 公司以及经理 Michael Marth 和 Philipp Suter 为本书的写作开绿灯。

OAuth 是一个被广泛应用的协议，由 IETF 旗下的众多同仁协作而成。他们中的有些人是安全领域的顶级专家。在写作过程中，我们有幸得到了 John Bradley、Hannes Tschofenig 和 William Dennis 的宝贵意见。

友谊会对一个人的生活产生不可思议的影响。因此，我要特别感谢这些人（排名不分先后）：Elia Florio 给了我源源不断的灵感；Damien Antipa 非常耐心地解释了 JavaScript 和 CSS 中最晦涩难懂的部分；Francesco Mari 带我进入了 Node.js 的缤纷世界，还不厌其烦地倾听我无尽的抱怨；Joel Richard 带我领略了 Apache Cordova 的魔力；Alexis Tessier 是我见过的最有才华的设计师；还有 Ian Boston 校对了本书。

最重要的，要感谢 Justin Richer，他是我所一直期望的理想合著者。Justin，你太棒了！

还要特别感谢我所爱的人们，没有你们我就无法完成本书。

感谢我的父母。他们总是鼓励我不断学习，并且不会给我任何压力，尽管他们自己没有学习的机会。他们的支持是独一无二的。还要感谢我的哥哥和姐姐对我的鼓励，特别是我刚上大学的时候。

当然，最要感谢的是我的未婚妻 Yolanda（马上就要结婚了），她不断鼓励我，并支持我所做的一切。最后，要感谢我的儿子 Santiago，他让生活中的每一天都很美好。我爱你。

关于本书

本书意在对 OAuth 2.0 以及包括 OpenID Connect 和 JOSE/JWT 在内的众多相关技术进行全面且透彻的探讨。希望读者读完本书之后，能对 OAuth 2.0 有深刻的理解，明白它的工作原理，还知道如何正确、安全地将其部署在并不安全的互联网上。

本书的目标读者可能使用过 OAuth 2.0，或者至少听说过它，但并不明白其工作原理。读者可能曾经开发过一些 OAuth 2.0 组件，比如与特定 API 交互的客户端，但也对其他类型的客户端或者 OAuth 2.0 生态系统中的其他部分充满好奇。读者可能想知道："当请求授权码时，授权服务器到底做了些什么？"或者，读者接到任务要对一个 API 进行保护，想确认 OAuth 2.0 是否能处理这个任务，如果它可以，又应如何驾驭它。也许读者的日常工作是开发客户端，但很想知道受保护的资源是如何处理发送过去的令牌的。又或者，读者正在构建一个受保护的 API，但想知道正在与其打交道的授权服务器是如何正确地发放令牌的。我们希望读者了解 OAuth 2.0 这个工具真正好在哪里，并且能有效地运用它。

我们假设读者了解基本的 HTTP 工作原理，至少理解 TLS 加密链接的作用，若了解其原理细节就再好不过了。本书使用 JavaScript，但并不讲解 JavaScript 的用法，我们会尽量解释代码所表达的抽象概念和功能本身，以便读者能将它应用到自己的平台和语言上。

路线图

本书分为 4 个部分，总共 16 章。第一部分由第 1~2 章构成，概述了 OAuth 2.0 协议，可以说是核心阅读材料。第二部分由第 3~6 章构成，展示了如何构建一个完整的 OAuth 2.0 生态系统。第三部分由第 7~10 章构成，讨论了 OAuth 2.0 生态系统中各个部分可能出现的漏洞，以及如何规避。最后一部分由第 11~16 章构成，这一部分跳出 OAuth 2.0 协议的核心部分，探讨更外围生态系统中的标准和规范，最后还对全书进行了总结。

- 第 1 章概述了 OAuth 2.0，讲述了开发它的动机，还介绍了 OAuth 出现之前与 API 安全相关的方法。
- 第 2 章深入讲解授权码许可类型，这是 OAuth 2.0 核心中最常用、最典型的一种许可类型。
- 第 3~5 章分别展示如何构建简易但功能完整的 OAuth 2.0 客户端、受保护的资源服务器，以及授权服务器。
- 第 6 章讨论 OAuth 2.0 协议内部的多样性，介绍了授权码之外的其他许可类型，还讨论了

原生应用中的许可类型。

- 第 7~9 章分别讨论 OAuth 2.0 客户端、受保护资源及授权服务器中常见的漏洞，以及如何避免这些漏洞。
- 第 10 章讨论 OAuth 2.0 中 bearer 令牌和授权码的弱点，针对它们的攻击，以及如何规避。
- 第 11 章介绍 JSON Web Token（JWT）及其所用的编码机制 JOSE，还包括令牌内省和撤回，这些主题完整覆盖了令牌的生命周期。
- 第 12 章介绍动态客户端注册，并讨论它对 OAuth 2.0 生态系统的影响。
- 第 13 章先解释为什么 OAuth 2.0 不是身份认证协议，继而介绍如何基于它使用 OpenID Connect 构建一个身份认证协议。
- 第 14 章介绍构建于 OAuth 2.0 之上的 User Managed Access（UMA）协议，该协议允许用户对用户（user-to-user）的分享。这一章还介绍了 HEART 和 iGov 这两个 OAuth 2.0 配置规范以及 OpenID Connect，以及这些协议在特定行业领域中是如何应用的。
- 第 15 章指出 OAuth 2.0 核心规范中的常规 bearer 令牌并不能满足所有需求，并描述了 Proof of Possession（PoP）令牌及 TLS 令牌绑定如何与 OAuth 2.0 协同工作。
- 第 16 章对全书进行总结，并指导读者如何进一步应用这些知识，还介绍了相关代码库以及范围更广的社区。

虽然我们按这样的次序组织编排了本书内容，但读者并非一定要按这样的顺序阅读。我们建议读者先阅读前两章，因为前两章对 OAuth 2.0 进行了全面概述，并深入介绍了关键概念和组件。不过说实话，读者可能在寻找某些特定的信息，所以可能会去阅读客户端开发和客户端弱点的相关章节，然后跳到用户身份认证或者令牌管理的章节，之后又去看与授权服务器相关的内容。因此，我们也试着确保让每一章相互独立，而且还对相关内容的引用提供了标注，以便读者查找。

代码

本书的代码采用 Apache 2.0 许可协议开源。虽然它们只是练习和示例，但我们也鼓励人们使用、重新组织以及贡献代码。像 OAuth 这样的开放标准与开源界是息息相关的，大家的贡献对它们来说非常重要。代码可以从 GitHub（https://github.com/oauthinaction/oauth-in-action-code/）获取，我们鼓励读者对其分叉、克隆、创建分支，甚至可以创建拉取请求来改进代码。第 3~13 章和第 15 章提供了实战代码，附录 A 对书中使用的框架进行了介绍，附录 B 列出了整理之后的代码清单。也可以从英文版出版社网站（https://www.manning.com/books/oauth-2-in-action）下载本书代码。

本书中的所有代码都使用运行于 Node.js 平台的 JavaScript 语言编写。书中大部分示例都是 Web 应用，使用了 Express.js 框架以及其他的库。我们已经尽最大努力让读者免受 JavaScript 特异性的困扰，因为本书的目标并不是仅让读者精通某一语言或平台。如果读者曾经使用过诸如 Java Spring 或者 Ruby on Rails 这样的 Web 框架，那一定对大部分概念和思想很熟悉。此外，本书还提供了实用函数，并附有文档说明，用于执行 OAuth 协议中的琐碎功能，比如构造包含查询参数的 URL，或者生成 HTTP 基本认证字符串。关于本书所使用的代码环境的更多细节，请参阅

附录 A，其中包含一个简单的练习，向读者展示了如何启动和运行代码。

还可以通过 Katacoda 在线运行本书的练习，Katacoda 是一个交互式的自学网站。这些练习使用的代码和本书中使用的完全相同，只是通过 Web 提供了一个容器化的运行环境。

代码约定

本书包含了大量示例源代码，它们要么被列在代码清单中，要么穿插在正文文本中。无论哪种情况，源代码都以 `fixed-width font like this` 这样的格式呈现，以区别于普通文本。有时候会以粗体来强调代码中相对于前一步骤的变化，比如向当前代码添加了新的功能。

在多数情况下，原始的源代码都经过了重新格式化；增添了换行符并调整了缩进，以适应图书页面上有限的空间。在极少数情况下，即使这样也还不够，还需要在代码清单中使用续行符号（➥）。另外，如果正文已对代码进行描述，源代码中的注释一般会被移除。很多代码清单中附有注解，以突出重要概念。

作者在线

购买了本书英文版就可以免费访问由 Manning 出版社运营的私密 Web 论坛，在这里读者可以发表对本书的评论，可以提出技术问题，并有可能得到作者或者其他用户的帮助。要访问和订阅论坛，可以使用浏览器打开 https://www.manning.com/books/oauth-2-in-action。该网页上说明了如何注册并进入论坛、可以获取哪些帮助以及论坛行为准则等。

Manning 出版社为读者和作者提供一个空间，让他们能够进行有意义的交流，但并不保证作者的参与程度，作者在论坛中的贡献都是自愿的（也是无偿的）。建议读者向作者多提一些具有挑战性的问题，这样会更引起他们的兴趣。

只要本书英文版处于销售状态，作者在线论坛以及上面的讨论就会一直可访问。

电子书

扫描如下二维码，即可购买本书电子版。

关于封面图片

本书封面上的画像题为“来自克罗地亚达尔马提亚扎格罗维奇的男子”。这张图片取自 19 世纪中期由 Nikola Arsenovic 绘制的克罗地亚传统服饰图集的复本，由克罗地亚斯普利特的 Ethnographic 博物馆于 2003 年出版。这些图片由斯普利特 Ethnographic 博物馆一位热心的管理员提供，该博物馆位于公元 304 年左右帝国皇帝 Diocletian 的宫殿遗址，这里曾是中世纪罗马帝国的中心。这本图集中有克罗地亚各个地区的图片，色彩斑斓，并附有对当地服饰和日常生活的介绍。

扎格罗维奇是达尔马提亚内陆的一个小镇，建在古老的中世纪城堡的遗址上。封面插图上的人物穿着蓝色羊毛长裤，在白色亚麻衬衣的外面，披着一件宽大的红色羊毛外套，服饰上布满了该地区特有的精致刺绣。他一手握着一根长烟斗，另一边肩上挂着一杆火枪，头戴红色帽子，脚穿鹿皮鞋。

在过去的 200 年里，着装规范和生活方式都发生了变化，当时地区之间的多样性已逐渐消失。现在，已经很难区分不同大陆的居民，更不用说只相隔几公里的不同村庄或城镇。也许，文化多样性已经转变成了更加多样化的个人生活——当然，是更加多样化和快节奏的科技生活。

Manning 出版社将反映两个世纪前各地区多彩生活的插图用作封面，来赞美计算机行业的活力和创新，也通过古老书籍和图册中的图片带我们领略过去的风土人情。

目　录

第一部分　起　步

第二部分　构建 OAuth 环境