

TURING

图灵程序设计丛书

*HTTP: The Definitive Guide*

# HTTP

权威指南



[美]

David Gourley, Brian Totty 著  
Reddy, Anshu Aggarwal

陈涓 赵振平 译

O'REILLY®

人民邮电出版社  
POSTS & TELECOM PRESS

**TURING** 图灵程序设计丛书

# HTTP权威指南

---

HTTP: The Definitive Guide

[美] David Gourley Brian Totty Marjorie Sayer  
Sailu Reddy Anshu Aggarwal 著  
陈涓 赵振平 译

**O'REILLY®**

*Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo*

l, Inc. 授权人民邮电出版社出版

人民邮电出版社  
北 京

---

# 版权声明

©2002 by O'Reilly Media, Inc.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and Posts & Telecom Press, 2012. Authorized translation of the English edition, 2012 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 O'Reilly Media, Inc. 出版 2002。

简体中文版由人民邮电出版社出版，2012。英文原版的翻译得到 O'Reilly Media, Inc. 的授权。此简体中文版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 的许可。

版权所有，未得书面许可，本书的任何部分和全部不得以任何形式重制。

---

# 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 的会议和峰会聚集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，而且有几次都是一闪即逝的机会，尽管大路也不错。”

——Linux Journal

---

# 前言

HTTP (Hypertext Transfer Protocol, 超文本传输协议<sup>①</sup>) 是在万维网上进行通信时所使用的协议方案。HTTP 有很多应用, 但最著名的是用于 Web 浏览器和 Web 服务器之间的双工通信。

HTTP 起初是一个简单的协议, 因此你可能会认为关于这个协议没有太多好说的。但现在, 你手上拿着的却是一本将近两斤重的书。如果你想知道我们会写出一本 700 多页的关于 HTTP 的书, 就去看看目录吧。本书不仅仅是一本 HTTP 首部参考手册, 它还是一本名副其实的 Web 架构“圣经”。

本书中, 我们会将 HTTP 中一些互相关联且常被误解的规则梳理清楚, 并编写了一系列基于各种主题的章节介绍 HTTP 各方面的特性。纵观全书, 我们对 HTTP “为什么” 这样做进行了详细的解释, 而不仅仅停留在它是“怎么做”的。而且, 为了节省大家寻找参考文献的时间, 我们还介绍了很多 HTTP 应用程序正常工作所必需且重要的非 HTTP 技术。在条理清晰的附录中, 可以找到按照字母排序的首部参考 (这些首部构成了最常见的 HTTP 文本的基础)。我们希望这种概念性的设计有助于读者更好地使用 HTTP。

本书是为所有希望理解 HTTP 和 Web 底层结构的人编写的。软硬件工程师也可以将本书作为 HTTP 及相关 Web 技术参考书使用。系统架构师和网络管理员可以通过本书更好地了解如何设计、实现并管理复杂的网络架构。性能工程师和分析人员可以从缓存和性能优化的相关章节中获益。市场营销和咨询专家还可以通过概念介绍更好地理解 Web 技术的前景。

---

<sup>①</sup> HTTP 译为“超文本传输协议”, 其中“transfer”使用了“传输”的含义, 但依据 HTTP 制定者之一 Roy Fielding 博士的论文, “transfer”表示的是“(状态的)转移”, 而不是“传输”。怎样翻译才更符合 HTTP 的原意, 其讨论可参见图灵社区的文章, 地址是 [ituring.com.cn/article/details/1817](http://ituring.com.cn/article/details/1817)。

本书澄清了一些常见的误解，推荐了“各种业内诀窍”，提供了便捷的参考资料，并且用通俗易懂的语言阐述了枯燥且令人费解的标准规范，还详细探讨了 Web 正常工作所必需且互相关联的技术。

本书创作历时良久，是由很多热衷于因特网技术的人共同完成的，希望它能对你有所帮助。

## 运行实例：Joe 的五金商店

本书的很多章节都涉及了一个假想的在线五金与家装商店示例，通过这个“Joe 的五金商店”来说明一些技术概念。我们为这个商店构建了一个真实的 Web 站点 (<http://www.joes-hardware.com>)，以便大家能够测试书中的部分实例。只要本书仍在销售，我们就会一直维护好这个 Web 站点。

## 本书内容

本书包含 21 章，分为 5 个逻辑部分（每部分都是一个技术专题），以及 8 个很有用的附录，这些附录包含了参考资料，以及对相关技术的介绍。

第一部分 HTTP：Web 的基础

第二部分 HTTP 结构

第三部分 识别、认证与安全

第四部分 实体、编码和国际化

第五部分 内容发布与分发

第六部分 附录

第一部分用 4 章的篇幅描述了 Web 的基础构件与 HTTP 的核心技术。

- 第 1 章简要介绍了 HTTP。
- 第 2 章详细阐述了统一资源定位符（Uniform Resource Locator, URL）的格式，以及 URL 在因特网上命名的各种类型的资源，还介绍了统一资源名（Uniform Resource Name, URN）的演变过程。
- 第 3 章详细介绍了 HTTP 报文是如何传送 Web 内容的。
- 第 4 章解释了 HTTP 连接管理过程中一些经常会引起误解且少有文档说明的规则和行为。

第二部分重点介绍了 Web 系统的结构构造块：HTTP 服务器、代理、缓存、网关以及机器人应用程序。（当然，Web 浏览器也是一种构造块，但在本书的第一部分已经对其进行过很详细的介绍了。）第二部分包含以下 6 章。

- 第 5 章简要介绍了 Web 服务器结构。
- 第 6 章深入研究了 HTTP 代理服务器，HTTP 代理服务器是作为 HTTP 服务与控制平台使用的中间服务器。
- 第 7 章深入研究了 Web 缓存的问题。缓存是通过保存常用文档的本地副本来提高性能、减少流量的设备。
- 第 8 章探讨了网关和应用服务器的概念，通过它们，HTTP 就可以与使用不同协议（包括 SSL 加密协议）的软件进行通信了。
- 第 9 章介绍了 Web 上的各种客户端类型，包括无处不在的浏览器、机器人和网络蜘蛛以及搜索引擎。
- 第 10 章讲述了仍在研究之中的 HTTP 协议：HTTP-NG 协议。

第三部分提供了一套用于追踪身份、增强安全性以及控制内容访问的技术和技巧。包含下列 4 章。

- 第 11 章讨论了一些识别用户的技术，以便向用户提供私人化的内容服务。
- 第 12 章重点介绍了一些验证用户身份的基本方式。这一章还对 HTTP 认证机制与数据库的接口问题进行了研究。
- 第 13 章详述了摘要认证，它是对 HTTP 的建议性综合增强措施，可以大幅度提高其安全性。
- 第 14 章说明了因特网的密码体系、数字证书以及 SSL。

第四部分涵盖 HTTP 报文主体和 Web 标准，前者包含实际内容，后者描述并处理主体内容。第四部分包含以下 3 章。

- 第 15 章介绍了 HTTP 内容的结构。
- 第 16 章探讨了一些 Web 标准，通过这些标准，全球范围内的用户都可以交换以不同语言和字符集表示的内容。
- 第 17 章解释了一些用于协商可接受内容的机制。

第五部分介绍了发布和传播 Web 内容的技巧。包括以下 4 章。

- 第 18 章讨论了在现代的网站托管环境中布署服务器的方式以及 HTTP 对虚拟网站托管的支持。
- 第 19 章探讨了一些创建 Web 内容，并将其装载到 Web 服务器中去的技术。
- 第 20 章介绍了能够将输入 Web 流量分散到一组服务器上去的一些工具和技术。
- 第 21 章介绍了一些日志格式和常见问题。

第六部分是一些很有用的参考附录，以及相关技术的教程。

- 附录 A 详述了统一资源描述符 (Uniform Resource Identifier, URI) 方案所支持的协议。

- 附录 B 列出了 HTTP 的响应代码，方便使用。
- 附录 C 提供了 HTTP 首部字段的参考列表。
- 附录 D 列出了大量的 MIME 类型，解释了 MIME 类型的注册方式。
- 附录 E 介绍了 HTTP 认证中使用的 Base-64 编码。
- 附录 F 详述了如何实现 HTTP 中的各种认证方案。
- 附录 G 定义了 HTTP 首部的语言标签值。
- 附录 H 列出了用以支持国际化 HTTP 的字符编码。

每章都包含很多实例，以及到其他相关的参考资料的链接。

## 排版约定

本书使用了下列排版约定。

- 楷体

用于 URL、C 函数、命令名、MIME 类型、新术语的定义以及重点内容。

- 等宽字体

用于计算机的输出、代码以及所有文字文本。

- 加粗等宽字体

用于用户的输入。

## 意见及问题

请将有关此书的意见及问题发给出版商：

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

本书有一个 Web 页面，上面列出了勘误表、一些实例以及所有的附加信息。可以通过以下链接来访问这个页面。



<http://www.oreilly.com/catalog/htptdg/>

为本书提意见或者询问一些技术性问题，可以向以下地址发送邮件。

[bookquestions@oreilly.com](mailto:bookquestions@oreilly.com)

更多与书籍、会议、资源中心以及 O'Reilly 网络有关的问题，都请参见 O'Reilly 的网站。

<http://www.oreilly.com>

## 致谢

本书是很多人共同劳动的成果。五位作者要感谢一些人，感谢他们为这本书所作出的巨大贡献。

首先，我们要感谢 O'Reilly 的编辑 Linda Mui。Linda 早在 1996 年就与 David 和 Brian 进行了首次接触，她还提炼了几个概念，并将其融入到今天大家拿到的这本书中。Linda 还帮助我们这帮首次写书、徘徊不定的人协调一致地按计划逐步完成了这本书的写作（尽管我们完成得并不怎么快）。最重要的是，Linda 给了我们一个创作此书的机会。我们要对她表示由衷地感谢。

我们还要感谢以下人士，他们非常聪明博学而且非常友善，为校对、注释并修订本书草稿花费了大量精力。他们是：Tony Bourke、Sean Burke、Mike Chowla、Shernaz Daver、Fred Douglass、Paula Ferguson、Vikas Jha、Yves Lafon、Peter Mattis、Chuck Neerdaels、Luis Tavera、Duane Wessels、Dave Wu 和 Marco Zaghera。他们的一些观点和建议大大提升了本书的质量。

本书大部分精美的插图都是由 O'Reilly 的 Rob Romano 制作的。为了能够更加清晰地描述一些微妙的概念，本书使用了大量详尽备至的插图。其中很多插图制作起来都很费劲，而且还经过了大量的修改。如果一幅图相当于一千个字的话，Rob 就相当于为本书增加了数百页的篇幅。

Brian 还要特别感谢所有作者对本项目的付出。为了对 HTTP 作出首次详细而又切实可行的剖析，作者们投入了大量的时间。其间虽然出现了婚礼、孩子出世、刻不容缓的工作项目、创业公司起步以及就读研究生院等诸多问题，但作者们的共同努力使这个项目得以圆满完成。我们相信，每个人的努力付出都是值得的，而且最重要的是，这项工作为大家提供了一项很有价值的服务。Brian 还要感谢 Inktomi 的员工们，感谢他们的热情和支持，以及他们对 HTTP 在实际应用程序中应用状况的深

刻洞察力。同时，还要感谢 Cajun-shop.com 允许我们使用他们的站点来展示书中的一些范例。

David 要感谢他的家人，尤其是母亲和祖父长期以来不懈的支持。他要感谢那些在写书这几年中忍受他古怪作息习惯的家人们。他要对 Slurp、Orctomi 和 Norma 所做的一切表示感谢，还要感谢合作者们的辛勤工作。最后，他要感谢 Brian 说服自己再次冒险。

Marjorie 要感谢她丈夫 Alan Liu 的技术洞察力，以及他对家庭的支持和理解。Marjorie 还要感谢合作者们丰富且深刻的灵感和洞察力。在编写过程中能够与他们共同工作，她感到非常开心。

Sailu 要感谢 David 和 Brian 为他提供机会参与编写本书，感谢 Chuck Neerdaels 将他引入了 HTTP 的世界。

Anshu 要感谢他的妻子 Rashi 和他的父母。尽管本书的编写旷日持久，但家人依旧对他有着足够的耐心，不断地支持并鼓励他。

最后，作者们要集体感谢各位著名和无名的因特网先驱们，他们在过去 40 年中所做的研究、开发和传播工作对我们的科学界、社会及经济团体作出了巨大的贡献。没有他们的工作，就没有本书所要讨论的话题。

# 目录

## 第一部分 HTTP: Web 的基础

第 1 章 HTTP 概述	3
1.1 HTTP——因特网的多媒体信使	4
1.2 Web 客户端和服务端	4
1.3 资源	5
1.3.1 媒体类型	6
1.3.2 URI	7
1.3.3 URL	7
1.3.4 URN	8
1.4 事务	9
1.4.1 方法	9
1.4.2 状态码	10
1.4.3 Web 页面中可以包含多个对象	10
1.5 报文	11
1.6 连接	13
1.6.1 TCP/IP	13
1.6.2 连接、IP 地址及端口号	14
1.6.3 使用 Telnet 实例	16
1.7 协议版本	18
1.8 Web 的结构组件	19
1.8.1 代理	19
1.8.2 缓存	20
1.8.3 网关	20
1.8.4 隧道	21

1.8.5 Agent 代理	21
1.9 起始部分的结束语	22
1.10 更多信息	22
1.10.1 HTTP 协议信息	22
1.10.2 历史透视	23
1.10.3 其他万维网信息	23
<b>第 2 章 URL 与资源</b>	<b>25</b>
2.1 浏览因特网资源	26
2.2 URL 的语法	28
2.2.1 方案——使用什么协议	29
2.2.2 主机与端口	30
2.2.3 用户名和密码	30
2.2.4 路径	31
2.2.5 参数	31
2.2.6 查询字符串	32
2.2.7 片段	33
2.3 URL 快捷方式	34
2.3.1 相对 URL	34
2.3.2 自动扩展 URL	37
2.4 各种令人头疼的字符	38
2.4.1 URL 字符集	38
2.4.2 编码机制	38
2.4.3 字符限制	39
2.4.4 另外一点说明	40
2.5 方案的世界	40
2.6 未来展望	42
2.7 更多信息	44
<b>第 3 章 HTTP 报文</b>	<b>45</b>
3.1 报文流	46
3.1.1 报文流入源端服务器	46
3.1.2 报文向下游流动	47
3.2 报文的组成部分	47
3.2.1 报文的语法	48
3.2.2 起始行	50
3.2.3 首部	53
3.2.4 实体的主体部分	55
3.2.5 版本 0.9 的报文	55
3.3 方法	56
3.3.1 安全方法	56

3.3.2	GET	56
3.3.3	HEAD	57
3.3.4	PUT	57
3.3.5	POST	58
3.3.6	TRACE	58
3.3.7	OPTIONS	60
3.3.8	DELETE	60
3.3.9	扩展方法	61
3.4	状态码	62
3.4.1	100 ~ 199——信息性状态码	62
3.4.2	200 ~ 299——成功状态码	63
3.4.3	300 ~ 399——重定向状态码	64
3.4.4	400 ~ 499——客户端错误状态码	68
3.4.5	500 ~ 599——服务器错误状态码	69
3.5	首部	70
3.5.1	通用首部	71
3.5.2	请求首部	72
3.5.3	响应首部	74
3.5.4	实体首部	75
3.6	更多信息	77
<b>第 4 章</b>	<b>连接管理</b>	<b>79</b>
4.1	TCP 连接	80
4.1.1	TCP 的可靠数据管道	80
4.1.2	TCP 流是分段的、由 IP 分组传送	81
4.1.3	保持 TCP 连接的正确运行	82
4.1.4	用 TCP 套接字编程	84
4.2	对 TCP 性能的考虑	85
4.2.1	HTTP 事务的时延	86
4.2.2	性能聚焦区域	87
4.2.3	TCP 连接的握手时延	87
4.2.4	延迟确认	88
4.2.5	TCP 慢启动	89
4.2.6	Nagle 算法与 TCP_NODELAY	89
4.2.7	TIME_WAIT 累积与端口耗尽	90
4.3	HTTP 连接的处理	91
4.3.1	常被误解的 Connection 首部	91
4.3.2	串行事务处理时延	92
4.4	并行连接	94
4.4.1	并行连接可能会提高页面的加载速度	94
4.4.2	并行连接不一定更快	95

4.4.3 并行连接可能让人“感觉”更快一些 .....	95
4.5 持久连接 .....	96
4.5.1 持久以及并行连接 .....	96
4.5.2 HTTP/1.0+ keep-alive 连接 .....	97
4.5.3 Keep-Alive 操作 .....	98
4.5.4 Keep-Alive 选项 .....	98
4.5.5 Keep-Alive 连接的限制和规则 .....	99
4.5.6 Keep-Alive 和哑代理 .....	100
4.5.7 插入 Proxy-Connection .....	102
4.5.8 HTTP/1.1 持久连接 .....	104
4.5.9 持久连接的限制和规则 .....	104
4.6 管道化连接 .....	105
4.7 关闭连接的奥秘 .....	106
4.7.1 “任意”解除连接 .....	106
4.7.2 Content-Length 及截尾操作 .....	107
4.7.3 连接关闭容限、重试以及幂等性 .....	107
4.7.4 正常关闭连接 .....	108
4.8 更多信息 .....	110
4.8.1 HTTP 连接 .....	110
4.8.2 HTTP 性能问题 .....	110
4.8.3 TCP/IP .....	111

## 第二部分 HTTP 结构

<b>第 5 章 Web 服务器</b> .....	<b>115</b>
5.1 各种形状和尺寸的 Web 服务器 .....	116
5.1.1 Web 服务器的实现 .....	116
5.1.2 通用软件 Web 服务器 .....	117
5.1.3 Web 服务器设备 .....	117
5.1.4 嵌入式 Web 服务器 .....	118
5.2 最小的 Perl Web 服务器 .....	118
5.3 实际的 Web 服务器会做些什么 .....	120
5.4 第一步——接受客户端连接 .....	121
5.4.1 处理新连接 .....	121
5.4.2 客户端主机名识别 .....	122
5.4.3 通过 ident 确定客户端用户 .....	122
5.5 第二步——接收请求报文 .....	123
5.5.1 报文的内部表示法 .....	124
5.5.2 连接的输入/输出处理结构 .....	125
5.6 第三步——处理请求 .....	126

5.7	第四步——对资源的映射及访问	126
5.7.1	docroot	127
5.7.2	目录列表	129
5.7.3	动态内容资源的映射	130
5.7.4	服务器端包含项	131
5.7.5	访问控制	131
5.8	第五步——构建响应	131
5.8.1	响应实体	131
5.8.2	MIME 类型	132
5.8.3	重定向	133
5.9	第六步——发送响应	134
5.10	第七步——记录日志	134
5.11	更多信息	134
<b>第 6 章</b>	<b>代理</b>	<b>135</b>
6.1	Web 的中间实体	136
6.1.1	私有和共享代理	136
6.1.2	代理与网关的对比	137
6.2	为什么使用代理	138
6.3	代理会去往何处	143
6.3.1	代理服务器的部署	144
6.3.2	代理的层次结构	144
6.3.3	代理是如何获取流量的	147
6.4	客户端的代理设置	148
6.4.1	客户端的代理配置：手工配置	149
6.4.2	客户端代理配置：PAC 文件	149
6.4.3	客户端代理配置：WPAD	150
6.5	与代理请求有关的一些棘手问题	151
6.5.1	代理 URI 与服务器 URI 的不同	151
6.5.2	与虚拟主机一样的问题	152
6.5.3	拦截代理会收到部分 URI	153
6.5.4	代理既可以处理代理请求，也可以处理服务器请求	154
6.5.5	转发过程中对 URI 的修改	154
6.5.6	URI 的客户端自动扩展和主机名解析	155
6.5.7	没有代理时 URI 的解析	155
6.5.8	有显式代理时 URI 的解析	156
6.5.9	有拦截代理时 URI 的解析	157
6.6	追踪报文	158
6.6.1	Via 首部	158
6.6.2	TRACE 方法	162
6.7	代理认证	164

6.8	代理的互操作性	165
6.8.1	处理代理不支持的首部和方法	166
6.8.2	OPTIONS: 发现对可选特性的支持	166
6.8.3	Allow 首部	167
6.9	更多信息	167
<b>第7章</b>	<b>缓存</b>	<b>169</b>
7.1	冗余的数据传输	170
7.2	带宽瓶颈	170
7.3	瞬间拥塞	171
7.4	距离时延	172
7.5	命中和未命中的	173
7.5.1	再验证	173
7.5.2	命中率	175
7.5.3	字节命中率	176
7.5.4	区分命中和未命中的情况	176
7.6	缓存的拓扑结构	177
7.6.1	私有缓存	177
7.6.2	公有代理缓存	177
7.6.3	代理缓存的层次结构	179
7.6.4	网状缓存、内容路由以及对等缓存	180
7.7	缓存的处理步骤	181
7.7.1	第一步——接收	181
7.7.2	第二步——解析	182
7.7.3	第三步——查找	182
7.7.4	第四步——新鲜度检测	182
7.7.5	第五步——创建响应	182
7.7.6	第六步——发送	183
7.7.7	第七步——日志	183
7.7.8	缓存处理流程图	183
7.8	保持副本的新鲜	183
7.8.1	文档过期	184
7.8.2	过期日期和使用期	185
7.8.3	服务器再验证	185
7.8.4	用条件方法进行再验证	186
7.8.5	If-Modified-Since:Date 再验证	187
7.8.6	If-None-Match: 实体标签再验证	189
7.8.7	强弱验证器	190
7.8.8	什么时候应该使用实体标签和最近修改日期	190
7.9	控制缓存的能力	191
7.9.1	no-Store 与 no-Cache 响应首部	191



7.9.2	max-age 响应首部	192
7.9.3	Expires 响应首部	192
7.9.4	must-revalidate 响应首部	192
7.9.5	试探性过期	193
7.9.6	客户端的新鲜度限制	194
7.9.7	注意事项	194
7.10	设置缓存控制	195
7.10.1	控制 Apache 的 HTTP 首部	195
7.10.2	通过 HTTP-EQUIV 控制 HTML 缓存	196
7.11	详细算法	197
7.11.1	使用期和新鲜生存期	198
7.11.2	使用期的计算	198
7.11.3	完整的使用期计算算法	201
7.11.4	新鲜生存期计算	202
7.11.5	完整的服务器——新鲜度算法	202
7.12	缓存和广告	204
7.12.1	发布广告者的两难处境	204
7.12.2	发布者的响应	204
7.12.3	日志迁移	205
7.12.4	命中计数和使用限制	205
7.13	更多信息	205
<b>第 8 章</b>	<b>集成点：网关、隧道及中继</b>	<b>207</b>
8.1	网关	208
8.2	协议网关	210
8.2.1	HTTP/*：服务器端 Web 网关	211
8.2.2	HTTP/HTTPS：服务器端安全网关	212
8.2.3	HTTPS/HTTP 客户端安全加速器网关	212
8.3	资源网关	213
8.3.1	CGI	215
8.3.2	服务器扩展 API	215
8.4	应用程序接口和 Web 服务	216
8.5	隧道	217
8.5.1	用 CONNECT 建立 HTTP 隧道	217
8.5.2	数据隧道、定时及连接管理	219
8.5.3	SSL 隧道	219
8.5.4	SSL 隧道与 HTTP/HTTPS 网关的对比	220
8.5.5	隧道认证	221
8.5.6	隧道的安全性考虑	221
8.6	中继	222
8.7	更多信息	224