

Java Network Programming

第4版



Java网络编程

O'REILLY®
中国电力出版社

Elliote Rusty Harold 著
李帅 荆涛 等译

014059451

TP312JA
1610

第四版

Java网络编程

Elliote Rusty Harold 著
李帅 荆涛 等译



TP312JA
1610

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

O'Reilly Media, Inc. 授权中国电力出版社出版

中国电力出版社



北航

C1745723

图书在版编目 (CIP) 数据

Java网络编程: 第4版/ (美) 哈诺德 (Harold, E. R.) 著; 李帅等译. —北京: 中国电力出版社, 2014.9

书名原文: Java network programming, fourth edition

ISBN 978-7-5123-6188-1

I. ①J… II. ①哈… ②李… III. ①JAVA语言—程序设计 IV. ①TP312

中国版本图书馆CIP数据核字 (2014) 第151539号

北京市版权局著作权合同登记

图字: 01-2014-4091号

©2013 by O'Reilly Media, Inc.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and China Electric Power Press, 2014.
Authorized translation of the English edition, 2013 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2013。

简体中文版由中国电力出版社出版2014。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。

封面设计/ Karen Montgomery, 张健

出版发行/ 中国电力出版社 (<http://www.cepp.sgcc.com.cn>)

地 址/ 北京市东城区北京站西街19号 (邮政编码100005)

经 销/ 全国新华书店

印 刷/ 北京丰源印刷厂

开 本/ 787毫米×980毫米 16开本 29印张 545千字

版 次/ 2014年9月第一版 2014年9月第一次印刷

印 数/ 0001—3000册

定 价/ 78.00元 (册)

敬告读者

本书封底贴有防伪标签, 刮开涂层可查询真伪
本书如有印装质量问题, 我社发行部负责退换

版权专有 翻印必究

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，而且有几次都是一闪即逝的机会，尽管大路也不错。”

——Linux Journal

谨以此书献给我小狗Thor。

作者介绍

Elliott出生于路易斯安那州的新奥尔良，如今他还定期返回那里研究他的那盆秋葵。不过，他现在与他的妻子Beth、他的狗Thor一起，居住在布鲁克林附近的Prospect Height。他经常在行业会议上演讲，包括Software Development、Dr. Dobb's Architecture Design World、JavaZone、JAOO、SD Best Practices、Extreme Markup Languages和很多用户组。他的开源项目包括利用Java处理XML的XOM库，以及Amateur媒体播放器。

封面介绍

本书的封面动物是一只北美河獭（*Lutra canadensis*）。这些小的食肉动物会在美国和加拿大的水域出现，除苔地和美国西南部的闷热干燥区域不适合它们外，它们的栖息地几乎遍布各地。北美河獭大约重20磅，大概2.5英尺长，雌性一般比雄性长三分之一。它们的食物主要由鱼和青蛙等水生动物组成，但由于它们有三分之二的的时间生活在陆地上，所以偶尔也会吃鸟类或啮齿动物。它们的毛分为两层——粗糙的外层毛和厚密的内层毛，可以有效地防御寒冷，事实上，它们看起来很喜欢在冰雪中玩耍。当潜水时，河獭的脉搏会从正常的每分钟170下降到20，这样就能保存氧气，可以在水下待得时间更长。这种动物喜爱群居，易于驯养。在欧洲，人们就曾训练一种河獭来捕鱼供人们食用。

目录

前言	1
第1章 基本网络概念	11
网络	12
网络的分层	13
IP、TCP和UDP	18
Internet	22
客户/服务器模型	26
Internet标准	27
第2章 流	33
输出流	34
输入流	38
过滤器流	42
阅读器和书写器	50
第3章 线程	58
运行线程	60
从线程返回信息	64
同步	74
死锁	80
线程调度	81
线程池和Executor	92

第4章 Internet地址	95
InetAddress类	97
Inet4Address和Inet6Address	108
NetworkInterface类	109
一些有用的程序	111
第5章 URL和URI	118
URI	118
URL类	124
URI类	141
x-www-form-urlencoded	149
代理	154
通过GET与服务器端程序通信	157
访问口令保护的网站	160
第6章 HTTP	167
HTTP协议	167
HTTP方法	176
请求主体	178
Cookie	180
第7章 URLConnection	186
打开URLConnection	187
读取服务器的数据	188
读取首部	189
缓存	198
配置连接	206
配置客户端请求HTTP首部	213
向服务器写入数据	215
URLConnection的安全考虑	220
猜测MIME媒体类型	221
HttpURLConnection	222

第8章 客户端Socket	234
使用Socket	234
用Telnet研究协议	235
构造和连接Socket	247
设置Socket选项	255
Socket异常	263
GUI应用中的Socket	264
第9章 服务器Socket	277
使用ServerSocket	277
日志	290
构造服务器Socket	296
获得服务器Socket的有关信息	298
Socket选项	299
HTTP服务器	303
第10章 安全Socket.....	318
保护通信	319
创建安全客户端Socket.....	321
选择密码组	324
事件处理器	328
会话管理	329
客户端模式	330
创建安全服务器Socket	331
配置SSLServerSocket	335
第11章 非阻塞I/O	338
一个示例客户端	340
一个示例服务器	343
缓冲区	350
通道	367
就绪选择	377

第12章 UDP	381
UDP协议	381
UDP客户端	383
UDP服务器	385
DatagramPacket类	387
DatagramSocket类	396
一些有用的应用程序	408
DatagramChannel	418
第13章 IP组播	428
组播	429
使用组播Socket	438
两个简单示例	444

前言

Java 在过去20年中的发展着实惊人。既然Java的快速成长得益于Internet迅猛的发展，如果还认为利用Java进行网络编程对许多人来说仍然十分神秘，这就实在让人很难相信了。事情并不是这样。事实上，如本书所言，利用Java编写网络程序非常简单。过去在UNIX、Windows或Macintosh环境下有过网络编程经验的读者会惊喜地发现，利用Java编写同样的程序将会更为简单。Java核心API包括了完善的接口，可以提供大多数网络特性。实际上，对于用C或C++编写的应用层网络软件，用Java编写只会更简单，而不会更难。《Java网络编程（第四版）》将尽力向你展示如何利用Java的网络类库，快速而轻松地编写程序来完成一些常见的网络任务。这些任务包括：

- 通过HTTP浏览Web。
- 编写多线程服务器。
- 对通信进行加密，以保证机密性、真实性和消息完整性。
- 设计GUI客户端提供网络服务。
- 向服务器端程序提交数据。
- 使用DNS查找主机。
- 通过匿名FTP下载文件。
- 连接Socket完成底层网络通信。
- 组播到网络上的多个主机。

Java是第一个（不过不再是唯一的一个）提供了如此强大跨平台网络库的语言，可以处理所有这些任务。本书将向你展现这个函数库的强大能力，同时也会指出其复杂性。本

书的目的是使你能够将Java作为一个平台来完成重要的网络编程。为此，本书提供了网络基础的一般背景，并详细讨论了Java对于编写网络程序所提供的便利。你将学习到如何编写Java程序在Internet上共享数据，来实现游戏、协作、软件更新、文件传输等诸多方面。你还将对HTTP、SMTP、TCP/IP及其他支持Internet和Web的协议有深入的了解，了解其底层原理。当读完本书时，你将会获得必要的知识和工具，完全可以自行创建能够充分利用Internet的下一代软件。

关于第四版

1996年，在本书第一版的第1章中，我用了大量篇幅来介绍我认为可以用Java实现的那种动态的、分布式网络应用程序。在之后写这本书的后续版本时，最让人激动的一点是，几乎我预言的所有应用程序都得以实现。程序员们使用Java来查询数据库服务器，监视Web页面，控制望远镜，管理多用户游戏等，这些都是通过使用Java的内在功能来访问Internet。不论是一般意义上的Java，还是特别地用Java进行网络编程，都已经远远超出了广告宣传范畴，而进入了真正的实际应用程序。

本书也走过了一段相当长的路。第四版会用更多的笔墨来介绍HTTP和REST。HTTP原先只是众多网络协议之一，现在几乎可以把“之一”去掉，而作为网络协议的代名词。可以看到，其他协议通常都基于HTTP构建，在网络栈中建立自己的层。

Java 6、7和8中的java.net和支持包还有很多其他的小调整和更新，第四版中也将涵盖这些内容。这一版中介绍的新类包括：CookieManager、CookiePolicy、CookieStore、HttpCookie、SwingWorker、Executor、ExecutorService、AsynchronousSocketChannel、AsynchronousServerSocketChannel等。另外在Java的最后3个版本中，还为现有的一些类增加了很多其他方法，这些将在相应的章节中讨论。我还重写了这本书的很大一部分，以反映相应的变化，这包括一般意义上Java编程的变化，以及特殊意义上网络编程方面的改变。希望你能发现第四版与上一版相比，对Java网络编程的论述更为充分，更为准确，作为这一领域的权威教程和参考书，能让读者更乐于接受，生命力也将更长久。

本书结构

第1章“基本网络概念”，将详细阐释关于网络和Internet如何工作，程序员需要了解哪些知识。这一章还会介绍Internet的底层协议，如TCP/IP和UDP/IP。

接下来的两章将重点介绍Java编程的两个部分，这两个内容对几乎所有网络程序都至关重要，却经常被误解和误用，这就是I/O和线程。第2章“流”，将探讨Java的典型I/O模型，而不是新的I/O API，这些模型不会很快被废弃，它们仍然是大多数客户应用程序中

处理输入和输出的首选方式。理解Java在一般情况下如何处理I/O，这是理解Java如何处理网络I/O这种特殊情况的先决条件。第3章“线程”，将探讨多线程和同步，并特别强调线程如何用于异步I/O和网络服务器。

有经验的Java程序员可以略读或跳过这两章。不过，对所有人而言，第4章“Internet地址”都是必须要阅读的。它将展示Java程序如何通过InetAddress类与域名系统（Domain Name System，DNS）进行交互，这是所有网络程序都需要用到的一个类。一旦读完这一章，你可以根据你的兴趣和需要直接跳到书中相关的部分继续阅读。

第5章“URL和URI”，将探讨Java的URL类，这是从多种网络服务器中下载信息和文件的一个抽象类，功能相当强大。基于URL类，你不必考虑服务器所用协议的细节，就能连接到网络服务器，并下载文件和文档。由此，你可以像访问HTTP服务器或读取本地硬盘上的文件一样，使用相同的代码来连接FTP服务器。你还将了解更新的URI类，相对于URL类，URI类与标准更一致，用于识别资源而非获取资源。

第6章“HTTP”将更深入地讨论HTTP，主要介绍REST、HTTP首部和cookie。第7章“URLConnection”将介绍如何使用URLConnection和HttpURLConnection类，从而不只是从Web服务器下载数据，还能上传文档和配置连接。

第8章到第10章将讨论Java用于网络访问的低层Socket类。第8章“客户端Socket”，介绍Java socket API，特别是Socket类。在此将展示如何编写与各种TCP服务器（包括whois、dict和HTTP服务器）交互的网络客户端。第9章“服务器Socket”，将展示如何使用ServerSocket类编写以上以及其他协议的服务器。最后，第10章“安全Socket”，将介绍如何使用安全Socket层（SSL）和Java安全Socket扩展（JSSE）来保护客户端-服务器通信。

第11章“非阻塞I/O”，涵盖了专门为网络服务器设计的新的I/O API。这些API能够让程序在试图读取或写入Socket前确定连接是否已经准备就绪。这就允许单个线程同时管理多个不同的连接，从而减轻虚拟机的工作负载。对于不需要同时打开多个连接的小服务器或小客户端来说，这些新的I/O API并没有太大帮助，但对于那些高吞吐量的服务器而言，却能提供显著的性能提升，使它们能够以网络所能提供的尽可能快的速度来传输网络能处理的尽可能多的数据。

第12章“UDP”，介绍用户数据报协议（User Datagram Protocol，UDP）及与之关联的DatagramPacket和DatagramSocket类，它们提供了快速、非可靠的通信。最后，第13章“IP组播”，展示了如何使用UDP同时与多个主机通信。

读者对象

本书假定读者熟悉Java语言和编程环境，还了解面向对象编程的一般概念。本书不是基础的语言教程。你应当完全熟悉Java的语法，而且编写过简单的应用程序。如果熟悉基本的Swing编程也很有好处，不过除了几个例子会用到Swing以外，一般情况下对此并没有要求。如果遇到某个网络编程问题需要更深入地理解（如线程和流），我也会在这本书中谈到，至少会做简单的介绍。

但是，本书不要求你有网络编程的经验。在这本书中，你会找到网络概念和网络应用程序开发的全面介绍。我不认为你能够随口说出很多网络编程的缩写词（TCP、UDP、SMTP等）。你将在这里学习需要了解的所有内容。

Java版本

在Java 1.0之后，Java的网络类较之于核心API的其他部分，发展得非常缓慢。与AWT或I/O相比，几乎没有任何改变，仅仅有一些增加。当然，所有网络程序都广泛应用了I/O类，而且一些程序还大量使用了GUI。在编写本书时，我们假设你至少使用Java 5.0以上的版本。一般情况下，我会直接使用Java 5的特性，如泛型和改进的for循环等，而不会做更多解释。

对于网络编程，Java 5和Java 6的差别并不大。这两个版本的大多数例子看起来都完全相同。如果某个方法或类是Java 6、7或8新增的，我们会在它的声明后面加一个注释来特别指出，如下所示：

```
public void setFixedLengthStreamingMode(long contentLength) // Java 7
```

Java 7稍稍有些“好高骛远”，不过，有些情况下Java 7的特性可能特别有用或者很方便，例如，由于本书篇幅有限，要减少示例的篇幅以便放在这本书里，try-with-resources和multicatch就很有帮助，所以我并不忌讳使用Java 7的特性，只是会特别指出使用了这样一些特性。

不过，总的来讲，Java的网络API自Java 1.0以来都相对稳定。Java 1.0之后的网络API很少被废弃，而且新增的特性也相对较少。即使在发布Java 8之后使用这本书应该也没有什么问题。但是，支持类中已经有越来越多新的API，特别是I/O，在Java 1.0之后它已经有3次重大修订。

关于示例

本书中介绍的大多数方法和类都会通过至少一个完整的运行程序来讲解，尽管这个程序

可能很简单。从我的经验来看，一个完整的运行程序对于展示方法的正确用法非常重要。如果没有一个实际的程序，就很容易深陷于众多术语当中，或者往往会掩盖作者自己也不清楚的知识点。Java API文档本身对于方法调用的描述通常过于简洁。在本书中，我会努力做到不过于啰唆也不过分简单。如果你对某个知识点已经很清楚，则完全可以跳过有关的说明。你不需要录入和运行这本书中的每一个示例，但是如果学习某个方法时遇到麻烦，则至少需要有一个运行示例才行。

每一章都包括至少一个（通常有多个）比较复杂的程序，会基于比较实际的设置来展示这一章中介绍的类和方法。它们通常会依赖这本书中没有讨论的Java特性。确实，在许多程序中，网络部件只是源代码的一小部分，通常也是不太难的部分。虽然如此，如果其他语言不像Java这样将网络摆在如此核心的地位上，要用那些语言来编写这些程序的话，绝不会像用Java编写那么简单。代码中的网络部分相当简单，这一点也反映出网络已经成为Java的一个核心特性，而并非程序本身要操心的琐事。本书中出现的所有示例程序都可以在线下载，通常这些示例会有所修改，还可能增加新内容。你可以从<http://www.cafeaulait.org/books/jnp4/>下载这些源代码。

我已经在Linux上测试过所有示例，在Windows和MacOS X上也测试了其中很多示例。这里给出的大多数示例都应当可以在支持Java 5或以后版本的其他平台、其他编译器和虚拟机上很好地工作。如果一个示例用Java 5或6无法编译，最常见的原因可能是其中使用了try-with-resources和multicatch。可以很容易地将这些例子重写为支持之前的Java版本，不过这样一来，代码会烦琐得多。

我确实很遗憾，由于篇幅限制，我不得不做出一些妥协。首先，我很少检查前置条件。大多数方法都假定会传入正确的数据，而且没有考虑null检查和保证代码质量的类似原则。另外，我把每个代码块的缩进减少为2个字符，每个延续行的缩进减少为4个字符，而没有采用Java的标准（分别缩进4个字符和8个字符）。希望这些缺陷不会影响到你。好的一方面是，这些让步确实让这一版比上一版简短得多（少了近百页）。

本书约定

正文为Minion Pro，常规字体（正如你现在所看到的）。

等宽印刷字体（Constant width）用于：

- 代码示例和片段。
- 在Java程序中出现的内容，包括关键字、操作符、数据类型、方法名、变量名、类名和接口名。

- 程序输出。
- 出现在HTML文档中的标记。

粗体等宽字体 (**Constant width Bold**) 用于:

- 要在屏幕上逐字输入的命令行和选项。

斜体字体 (*italic*) 用于:

- 新定义的术语。
- 路径名、文件名和程序名 (但是, 当程序名也是Java类名时, 将像其他类名那样显示为等宽字体)。
- 主机和域名 (*www.hpmor.com*)。
- URL (*http://www.cafeaulait.org/slides/*)。
- 其他书的标题 (Java I/O)。

提示: 表示一个提示、建议或一般说明。

警告: 表示一个警告或警示。

重要的代码段和完整的示例一般都单独作为一段, 例如:

```
Socket s = new Socket("java.oreilly.com", 80);
if (!s.getTcpNoDelay()) s.setTcpNoDelay(true);
```

当代码表示片段而非完整程序时, 这暗指已经有适当的import语句。例如, 在上面的代码段中, 就可以假定已经导入了java.net.Socket。

有些示例混杂了用户输入和程序输出。在这种情况下, 用户输入显示为粗体, 如第9章中的下面这个例子:

```
% telnet rama.poly.edu 7
Trying 128.238.10.212...
Connected to rama.poly.edu.
Escape character is '^]'.
This is a test
This is a test
This is another test
This is another test
9876543210
9876543210
```

```
^]
telnet> close
Connection closed.
```

最后，尽管这里使用的许多示例只是玩具示例，并且不具有重用性，但我开发的一些类仍具有实用价值。你可以在你自己的代码中随意使用这些类或者其中的任何部分，而不需要特殊的许可。它们是公开的（不过，这绝对不包括那些说明文字）。

意见反馈

我很乐于听到来自读者的声音，无论是关于本书的一般意见，特定的修正，希望涵盖的其他主题，或者只是你自己在网络编程斗争中的故事，都可以向`elharo@ibiblio.org`发送电子邮件与我联系。但是要知道，我每天都会收到几百封邮件，无法亲自一一回复。为尽可能地得到亲笔回复，请注明你是本书的读者。如果某个程序不能像你预期的那样运行，并对此感到疑惑不解，请尽量简化情况的说明，并能再次生成这个bug，最好是一个类，将整个程序的文本粘贴在你的电子邮件的正文中。未经请求的附件会直接删除而不会打开。另外，一定要用你希望得到回复的那个地址来发送邮件，还要确保正确地设置了你的Reply-on（回复）地址！如果我花费了一个小时或更长时间来寻找一个有趣问题的答案，而且写了一个详细的回复，却发现邮件被退回，而退回的原因只是因为与我通信的人是在公用终端上发送邮件，没有设置浏览器首选项来包括其实际电子邮件地址，再没有比这更让人沮丧的事了。

我还坚信一句老话：“如果你喜欢这本书，请告诉你的朋友。如果你不喜欢它，请告诉我。”我特别希望听到关于错误的信息。这是这本书的第四版。我已经竭尽所能让它成为一本完美的书，不过我还在努力。虽然我和O'Reilly公司的编辑们为本书奋力工作，但我确信肯定还有一些地方存在错误或排版问题。而且我相信至少还存在一个很棘手的大问题。如果你找到一个错误或排版问题，请告诉我，以便改正。我将把勘误发布在O'Reilly的Web网站上（http://oreil.ly/java_np_errata）。在向我报告错误之前，请检查这些页面，看看我是否已经知道了这个错误并且发布了勘误。所有已报告的错误都将在下一次印刷时改正。

使用代码示例

这本书将成为你工作的助手。一般说来，如果书中提供了代码示例，你可以在你的程序和文档中使用这些代码，除非复制使用了本书的大部分代码，否则不需要联系我们申请获得许可。例如，如果只是编写一个程序，其中用到了本书的几个代码段，这是不需要许可的。销售或发行O'Reilly图书的示例光盘则需要得到许可。如果引用本书的文字以