



Java Servlets 2.3

编程指南

〔美〕 John Bell 等著
Tony Loton

马树奇 等译

Professional Java Servlets 2.3

Java Servlets 2.3

编程指南

[美] John Bell 等著
Tony Loton

马树奇 等译

电子工业出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 提 要

本书旨在介绍如何使用Java Servlets技术生成功能强大并且可以移植的企业应用程序组件, 如何使用servlet来控制应用程序运行的流程、跟踪应用程序用户、截获及修改请求和响应信息, 以及如何与Web服务交互。

本书适用于熟悉Java语言及Java核心API的开发者, 是美国多位Java技术专家的经验总结与成果汇集。



Copyright©2002 Wrox Press. All rights reserved. No part of this book may be reproduced, stored in a retrieval system or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embodied in critical articles or reviews.

本书英文版由Wrox公司出版, Wrox公司已将中文版独家版权授予电子工业出版社及北京美迪亚电子信息有限公司。未经许可, 不得以任何形式和手段复制或抄袭本书内容。

版权贸易合同登记号: 01-2002-0170

图书在版编目(CIP)数据

Java Servlets 2.3编程指南/(美) 贝尔(Bell, J.)等著; 马树奇等译.-北京: 电子工业出版社, 2002.7

书名原文: Professional Java Servlets 2.3

ISBN 7-5053-7738-8

I. J... II. ①贝... ②马... III. Java语言—程序设计 IV. TP312

中国版本图书馆CIP数据核字(2002)第043643号

责任编辑: 李 莹

印 刷: 北京天竺颖华印刷厂

出版发行: 电子工业出版社 <http://www.phei.com.cn>

北京市海淀区万寿路173信箱 邮编: 100036

北京市海淀区翠微东里甲2号 邮编: 100036

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 37.75 字数: 970千字

版 次: 2002年7月第1版 2002年7月第1次印刷

定 价: 56.00元

凡购买电子工业出版社的图书, 如有缺损问题, 请向购买书店调换, 若书店售缺, 请与本社发行部联系。联系电话: (010) 68279077

序

欢迎学习专业Java Servlets 2.3。本书旨在介绍如何使用Java Servlets技术生成功能强大并且可以移植的企业应用程序组件，以及如何在Web应用程序中应用。

近些年来，人们通常把分布式业务服务中的交互活动放在Web层，并且使用servlet作为Web应用程序的动力引擎和组件框架。为此，我们将介绍如何使用servlet来控制应用程序运行的流程、跟踪应用程序用户、截获及修改请求和响应信息，以及与Web服务交互。

随着Servlet技术规范2.3版的开发（最终于2001年9月完成），专家组进行了一系列的修改，其中包括：

- 添加过滤处理
- 溶入侦听或者存活期事件（lifecycle）
- 使用J2SE 1.2作为Web容器的基层系统平台
- 在技术规范中采用Javadoc API定义
- 提高国际化处理水平
- 增强Java Archive（JAR）相关性
- 改善类装入器
- 还有其他许多修改，如一种新型经过改善的错误属性，以及会影响HTTPS请求的新的安全属性
- 引入新类和方法的一些变化，并且不建议使用javax.servlet.http.HttpUtils类

Servlets 2.3 API技术规范也是Java 2 Platform、Enterprise Edition（J2EE）1.3技术规范中的一个关键部分，在本书中你将会看到，它在J2EE系统平台的应用程序中作为控制器扮演着非常重要的角色。

随着内容的展开，我们将详细介绍这些概念，同时使用充分而完整的工作示例来说明它们的使用。

本书适合的对象

本书针对的是已经熟悉了Java语言及Java核心API的开发者。同时，本书假设读者已经熟悉了HTML和XML的一些基本知识，但这并不是基本要求。我们将使用最新版本的技术规范，这就是Java Servlets 2.3版。

Servlet很少独立使用，而本书也并不宣称这里就覆盖了所有领域的各种情况，特别是与其他Java技术及API如JDBC、JNDI和JavaServer Pages相关的领域。读者可以参考Wrox Press公司出版的《Professional Java Server Programming J2EE 1.3 Edition》（ISBN 1-861005-37-7），该书对于整个J2EE系统平台进行了出色的介绍。

本书介绍的内容

本书的结构如下：

- 第1章综述如何把servlet融合到企业应用中，以及该把它们用于什么场合。
- 第2章和第3章介绍Servlet 2.3 API。我们将学习servlet的存活期，并且理解如何使用及生成HTTP请求和响应信息。
- 尽管我们在前面的几章中也会运行一些应用程序，但第4章才介绍Web应用程序的结构以及如何在Web服务器上部署它们。
- 第5章～第7章介绍servlet的一些强大特性，包括如何维护会话、如何保持servlet以及什么是过滤器（filter）。
- 第8章介绍JavaServer Pages（Java服务器页面，JSP），这是一种对Java Servlets构成有益补充的技术。
- 第9章～第11章介绍在生产环境中部署Web应用程序时将会遇到的一些问题，以及跟踪servlet中出现问题的一些调试技术，同时了解如果不考虑类装入和同步处理的效果可能出现什么问题。
- 第12章和第13章介绍Web应用程序设计对系统性能和可维护性会产生怎样的影响。我们将研究各种不同的模式来生成更好的应用程序，介绍一些技术和工具，以便提高Web应用程序的性能和扩展能力。
- 最后一章也就是第14章介绍如何使用servlet作为代理程序访问来自Web服务（web service）的信息。

学习本书内容需要什么条件

本书中的大部分代码都已经用Java 2 SDK 1.3版（此SDK可以从网址<http://java.sun.com/j2se/1.3/>下载）和Apache Tomcat 4（可以从<http://jakarta.apache.org/tomcat/>下载）测试过。但是运行一些章节的示例的时候还需要额外一些软件。

本书中有几章的内容需要访问数据库。在这些章节中我们使用的是MySQL（版本3.23）以及MM.MySQL JDBC驱动程序（版本2.0.6）。可以从网址<http://www.mysql.com>下载上述两个软件。下载的文件中包含有完整的安装指导。

有几章还需要其他一些软件：

- Java 2 SDK 1.4版——可以从网址<http://java.sun.com/j2se/1.4/>下载
- The JavaBeans Activation Framework——可以从网址<http://java.sun.com/products/javabeans/glasgow/jaf.html>下载
- XML Parsers: 我们已经使用了Apache Xerces——可以从<http://xml.apache.org/xerces-j/index.html>下载，以及Apache Xalan XSLT处理器——可以从<http://xml.apache.org/xalan-j/>下载

- Apache SOAP 2.2——可以从<http://xml.apache.org/soap/index.html>下载
- Apache AXIS——可以从<http://xml.apache.org/axis/index.html>下载

本书中的代码可以在一台连网的计算机中运行（也就是说，可以通过本地浏览器从<http://localhost/>上查看）。

本书中的全部源代码可以从下列网址下载：

<http://www.wrox.com/>

约定

为了帮助读者从文本内容中获取更多的知识并且跟上内容的发展，我们在本书中使用了一系列的约定。

例如：

用黑体字标出重要的、不能忘记的信息，这些内容直接与周围的文本相关。

用这种楷体字辅助说明当前的讨论内容。

我们还以不同的方式显示代码。方法的定义和属性显示样式如下：

```
protected void doGet(HttpServletRequest req, HttpServletResponse resp)
    throws ServletException, IOException
```

示例代码的显示样式如下：

```
In our code examples, the code foreground style shows new, important,
pertinent code
while code background shows code that's less important in the present context,
or has been seen before.
```

客户支持

我们非常珍视来自读者的意见，希望了解你对本书有什么想法——喜欢什么、不喜欢什么、想的是什么——这些对我们今后的工作都会有好处。你可以把自己的意见通过电子邮件发送给feedback@wrox.com。请注意在电子邮件消息中列出本书的标题。

如何下载本书的示例代码

访问Wrox站点<http://www.wrox.com/>时，只需简单地通过Search（搜索）机制从书目列表中找出自己需要的名称。点击Code列中的Download链接或者在相关书目的详细信息页面上点击Download Code即可进行下载。

在我们的站点上可供下载的文件均经过WinZip进行了压缩。把附件存储在计算机硬盘上的文件夹中时，需要使用一个解压缩程序如WinZip或PKUnzip来展开这些文件。展开这些文件之后，各章的代码通常会展开到各章对应的文件夹中。当开始展开处理时，应保证自己

的软件（WinZip、PKUnzip等）使用了对应的文件夹名字。

勘误

我们已经尽了最大的努力来保证在内容或者代码中尽可能没有错误。但是人无完人，错误总难避免。如果你发现我们的书中有什么错误，如代码错误，请给予我们相关的反馈消息，我们将不胜感激。因为这些勘误信息足可以为其他读者节省数小时以解决可能遇到的困难。当然你也是在帮助我们提供更高质量的信息。只要给地址support@wrox.com发送电子邮件即可，你的信息将得到审核，如果正确，我们会登出相关页的勘误信息，或者在本书后续版本中使用。

如果想在Web站点中查看勘误信息，请进入网址<http://www.wrox.com/>，简单地通过我们的Advanced Search或者标题列表找出相关书目，点击位于相关书目详细页面封面图形下面的Book Errata链接即可找到。

电子邮件支持

如果你想直接就本书中的一些问题询问了解本书详细情况的专家，可以发电子邮件到support@wrox.com，列出书名以及ISBN中的最后4个编号。典型的电子邮件如下所示：

- 在Subject（主题）字段中列出书名、ISBN的后4位，有问题的页码。
- 你的姓名、联系信息以及问题消息正文。

我们不会给你发垃圾邮件。我们需要这些详细信息是为了节省你我双方的时间。当你发送一封电子邮件消息时，该消息会经过下列支持过程：

- 客户支持（Customer Support）——首先把消息送到我们的客户支持技术人员手中，他们拥有最经常问及情况的文件，将回答关于本书或者Web站点等直接问题。
- 编辑（Editorial）——比较深入的问题会转发给负责本书相关事务的技术编辑。他们具有编程经验或者关于特定产品的经验，能够回答相关主题的详细技术问题。
- 作者（Author）——最后，当出现了编辑无法回答的问题时（当然这种情况极少出现），他们会把相关请求转给作者。我们会保护作者在写作过程中不受任何打扰，但我们也很愿意把有价值的信息转给他们。Wrox公司的所有作者都会为其著作的技术支持提供帮助。他们将通过电子邮件与客户及编辑联系，发出响应信息，以使所有读者受益。

Wrox公司的技术支持过程只对直接与我们出版的书籍有关的问题做出解答。对于不属于此类范围的问题，可以从我们的<http://p2p.wrox.com/>论坛获取支持。

p2p.wrox.com

如果想与作者进行对等讨论可以加入P2P邮件群。我们这个系统提供了程序员到程序员（programmer to programmer™）之间的联系，除了我们的一对一电子邮件支持系统以外，可以通过邮件群、论坛和新闻组来实现。如果你向P2P发出一个查询，那么请相信此消息肯定会被许多Wrox作者及其他业界专家看到，他们都参与我们的邮件群。在p2p.wrox.com，有许多不同的邮件群可以提供帮助，不论你是在阅读本书，还是在开发自己的应用程序。特别针对

本书的邮件群是j2ee和pro_java_server。

如果想订阅这样的邮件群，可以按照下列步骤操作：

1. 进入<http://p2p.wrox.com/>
2. 从左侧菜单栏中选择适当的分类
3. 点击你想要加入的邮件群
4. 遵照其中的指示进行订阅，并且填写你自己的电子邮件地址和口令
5. 对收到的要求确认的电子邮件进行答复
6. 使用订阅管理器来加入更多的邮件群以及设置你自己的电子邮件首选项

我们会提供最好的支持

你可以选择加入邮件群或者接收他们的每周摘要。如果你没有时间、设施来接收邮件群，那么可以搜索我们的在线档案。垃圾邮件会被删除，你自己的电子邮件地址由专门的Lyris系统保护。关于加入或者离开邮件群的指示及其他一般邮件群查询，可以发电子邮件给listsupport@p2p.wrox.com。

作者简介

Subrahmanyam Allamaraju

Subrahmanyam是一位BEA Systems公司的高级工程师。由于他对建模有浓厚的兴趣，因此在获得了电气工程博士之后开始进入面向对象的编程技术领域，然后又深入到了分布式计算和软件体系结构的研究当中。在此过程中，他也从自己原来的家乡——Indian Institute of Technology（印度技术研究所）转到了Computervision，然后是Wipro Infotech，后来又到了BEA Systems公司。关于他目前从事的工作情况在他的个人主页上介绍得更多，其网址是<http://www.Subrahmanyam.com>。

Subrahmanyam对Varaa在时间很紧的情况下能够同舟共济并提供了示例代码表示感谢。

Subrahmanyam编写了本书的第5章。

John T. Bell

John Bell是Latitude360的首席工程师，Latitude360是RWD Technologies的一个附属机构。他有超过20年提供技术解决方案的经历，并且在过去几年中一直从事基于Java的Web系统的开发工作，非常熟悉面向对象的开发和分布式处理系统。最近他作为首席工程师对戴姆勒·克莱斯勒公司的许多Java Web系统提供了技术支持，现在正服务于Latitude360自己的eLearning产品开发项目。他获得了美国马里兰大学电气工程学士学位和计算机管理系统硕士学位。在业余时间里，他喜欢与孩子们共度时光、为当地的慈善机构维护计算机系统以及编写策略型计算机游戏，同时还积极参与教会组织的许多活动。

感谢Latitude360和RWD Technologies为我提供了时间、设备和必需的软件，使我能够完成这本书的编写工作。

献给我的父母。我的母亲赋予了我写作的天才，父亲把我引入了电子、计算机和工程技术领域。

John编写了本书的第14章。

Simon Brown

Simon是一位技术设计师（Technical Architect），工作在伦敦，从1996年起一直使用Java。在这段时间里他参与了许多Java开发工作，曾任技术主管、顾问及教员。工作之余他还出席许多Java专栏，其中包括JavaOne，曾经在JavaWorld发表文章，也是Wrox Press公司出版的《Professional JSP》第二版的作者之一。

Simon 1996年毕业于University of Reading，获得了计算机科学First Class BSc（Hons）学位。他还参加了Sun公司的许多认证，其中包括Sun Certified Enterprise Architect for J2EE

(Sun认证的J2EE企业系统设计师)、Web Component Developer for J2EE (J2EE Web组件开发人员)和Developer for the Java 2 Platform (Java 2系统平台开发人员)。如果您有任何问题或者意见, 欢迎与他通过电子邮件联系, 地址为projavaservlets@simongbrown.com。

我再次感谢未婚妻Kirstie对我的鼓励, 并且让我每次都连续数小时地在房间里独立工作。

Simon编写了本书的第12章。

Sam Dalton

Sam是一位技术设计师, 工作在伦敦。他有多年的Java及相关技术工作经验, 曾经服务于多种行业, 其中包括投资银行、保险及电子商务零售业。

Sam非常幸运地在2001年的JavaOne中发表及协作发表过多篇文章, 相信他在将来还会有更多的发展。

Sam毕业于University of Reading, 获得了计算机科学领域的荣誉学位, 现在是一位Sun公司认证的Java 2系统平台程序员、开发人员以及Web组件开发人员。

感谢我的妻子Anne。她总是给予我莫大的支持以及无微不至的关怀。

Sam编写了本书的第4章。

Andrew Harbourne-Thomas

Andrew Harbourne-Thomas是一位独立顾问人员, 专攻J2EE应用程序设计、开发以及项目管理。他刚开始工作时是一位经济学家, 曾任Irish Trade Board (爱尔兰贸易组织, 现在的Enterprise Ireland) 经济顾问, 后来多年从事世界顶级公司独立顾问的工作, 服务于微软等公司, 致力于技术、策略规划和项目管理。

他始终对技术发展有浓厚的兴趣, 从1997年开始从事Java技术的相关工作, 有时还在Bear Stearns IT分部工作。他感兴趣的领域包括J2EE应用程序体系结构、Web服务设计、极端编程和刚刚出现的新技术。

Andrew住在都柏林, 除了Java之外他还喜欢潜水和摄影。

感谢Miriam给我的爱、耐心和支持。

Andrew编写了本书的第1章至第3章及附录B。

Bjarki Holm

在过去的5年中, Bjarki就职于VYRE Corporation的软件工程部门, 位于雷克雅未克, 从事在Oracle后端系统上使用servlet进行嵌入式程序设计和内容管理系统的工作。目前他已经离职进修电气工程, 偶尔也为Wrox编写著作。

Bjarki编写了本书的第6章。

Meeraj Moidoo Kunnumpurath

Meeraj是EDS系统方面的一位高级信息专家（Senior Information Specialist）。他曾经使用J2EE和XML设计了企业桌面帮助系统和付款系统。

希望把我在本书中所做的工作献给我生命中最出色的三位女士：我的母亲、我的妻子和我的妹妹。愿安拉保佑全人类。

Meeraj编写了本书的第8章和第13章。

Sing Li

自从1978年受过计算机缺陷方面的冲击之后，Sing已经随着微处理器和因特网技术的发展而成长了起来。他的第一台PC机是花了99美元自己动手组装的COSMIC ELF计算机，只有256字节的内存和一位LED显示器。20年后，Sing已经成为一位多产的作者、顾问、教员、企业家和演讲者。他丰富的阅历包括了分布式体系结构、Web服务、多层服务器系统、计算机技术、通用消息处理和嵌入系统。Sing因最早著作了关于因特网全球电话（Internet Global Phone）的书籍而闻名，他通过IP技术进行了远程语音传送，这比现在出现的相关技术早了许多年。Sing在过去曾参与Wrox的多项开发项目，从事Java、Jini和JXTA方面的工作（及写作），并且着手于这些技术刚刚发布可用版本的最早时刻。他还积极倡导P2P系统技术，并持有该项技术拥有无限潜力的观点。

Sing编写了本书的第7章和第9章。

Tony Loton

Tony Loton是LOTONtech Limited公司（<http://www.lotontech.com>）的首席顾问/主管。

他在自己的公司中是一位独立顾问、培训教员和自由撰稿人，目前感兴趣的领域是LOTONtech，其中包括基于Java的语言分析和自动从万维网（World Wide Web）上提取信息。

Tony拥有计算机科学和管理方面的学士学位，拥有10年以上的IT工作经验，5年以上的Java、UML及相关技术经验。他的著作曾经在Wrox Press、John Wiley & Sons及许多IT业杂志中出版。

把我的成绩献给我的孩子Becky和Matt，我生命里的希望之光。

Tony编写了本书的第10章和第11章。

目 录

第1章 企业中的servlet	1
企业应用程序的结构	1
J2EE容器结构	6
J2EE Web组件	10
利用其他的J2EE API使用servlet	17
Web应用程序中的servlet角色	20
小结	28
第2章 Servlet 2.3 API	29
javax.servlet包	29
Servlet接口	32
GenericServlet类	35
请求 - 响应循环	40
输入和输出流	56
servlet - 容器通信	58
其他接口	68
servlet异常类	73
个人门户Web应用程序	74
小结	88
第3章 HTTP servlet	89
应用层协议	89
超文本传输协议	90
HTTP和servlet	93
实现HTTP servlet	102
servlet和自定义客户	110
小结	140
第4章 部署Web应用程序	141
什么是Web应用程序	141
Web应用程序的结构	142
ServletContext	145
Web应用程序的存活期	146

部署描述符	147
部署示例Web应用程序	161
高级部署问题	173
小结	174
第5章 会话处理	176
HTTP的无状态性	176
为什么要跟踪客户身份和状态	177
如何维护会话	178
使用Servlet API进行会话管理	189
小结	215
第6章 servlet持久性和资源	216
持久性资源	216
servlet初始化和配置	218
与JNDI绑定的资源	222
数据源绑定	225
访问持久性资源	232
servlet持久性	249
小结	262
第7章 过滤器	263
什么是过滤器	263
理解过滤器	264
配置过滤器	267
使用过滤器	269
链接过滤器	280
小结	282
第8章 Java服务器页面	283
JSP基础	283
JSP脚本元素	286
JSP隐含对象	287
指令	290
JSP操作	291
结合使用servlet和JSP	301
小结	317

第9章 安全性和容器的身份验证	319
服务器上的Java 2安全模型	319
安全套接层	323
Servlet 2.3安全性	332
使用声明性安全	338
程序式安全措施	351
小结	353
第10章 servlet调试技术	354
servlet调试问题	354
调试技术	355
用过滤器进行调试	356
用事件侦听器进行调试	359
用JPDA进行调试	363
观察调试跟踪	371
选择一种调试技术	373
J2SE 1.4中的日志记录增强	374
用UML进行运行时逆向分析处理	376
小结	377
第11章 类的装入和同步处理	378
类的装入	378
servlet容器中的类装入器	379
类的装入对应用程序逻辑产生的影响	384
servlet装入和重装	387
线程化和同步处理	387
小结	403
第12章 设计Web应用程序和servlet模式	404
良好的应用程序设计为什么很重要	404
J2EE Web应用程序设计	405
设计原则的文档记录	410
生成一个基于Web的论坛	411
使用Model 1结构建立论坛	416
重新考虑这个应用程序	425
使用Model 2结构建立论坛	426
小结	443

第13章 性能和扩展能力	445
13.1 良好的编程实践	445
13.2 程序结构方面的考虑	450
13.3 分析工具	475
13.4 自定义的性能监视工具	481
13.5 小结	488
第14章 Web服务和servlet代理	490
14.1 Web服务	490
14.2 Web服务的使用	492
14.3 Apache SOAP	496
14.4 servlet代理	499
14.5 整合与聚合	508
14.6 现在和未来的挑战	513
14.7 事务和Web服务	515
14.8 小结	516
附录A 安装Tomcat 4.0	518
附录B HTTP参考	527
附录C Servlet 2.3 API参考	554

第1章 企业中的servlet

Java是一种成熟的技术，可以按照不同的系统平台来划分。目前比较流行的是Java 2 Platform, Standard Edition (Java 2系统平台标准版, J2SE)，它提供了核心的Java API和Java Virtual Machine (Java虚拟机, JVM)，以及一些开发工具，如Java编译器等。但是本书中我们将关注Java 2 Platform, Enterprise Edition (Java 2系统平台企业版, J2EE)。实际上，我们只集中研究J2EE的一部分，这就是Java Servlet 2.3 API。

J2EE建立在J2SE的基础上，为开发和部署企业应用程序提供API和服务。将J2SE和J2EE的服务和库结合在一起有助于开发独立于系统平台、基于Web的Java应用程序。J2EE系统平台还有助于服务器厂商提供相关的环境来部署和运行J2EE应用程序。

- 在本章中，我们首先考虑分布式应用中的企业结构以及Web应用程序开发中涉及到的各层。
- 接着，观察J2EE结构怎样能够通过Web容器和Java servlet为Web开发提供便利，并且讨论servlet开发者希望在其应用程序中包含的J2EE服务和库。
- 讨论在Web应用程序中使用遵守Servlet 2.3技术规范的Java servlet会有哪些优点。
- 讨论容器厂商和开发者在实现Servlet技术规范时扮演的角色，介绍Tomcat Web容器/服务器，这两者是Servlet技术规范的参考实现。
- 在本章结束的时候，我们将观察在现代企业应用程序中servlet需要扮演的角色，以此说明它们在企业应用程序开发过程中的相关性和重要性。

但在我们详细介绍servlet之前，先来从较高的层面观察一下企业应用程序的结构。

企业应用程序的结构

当今企业应用程序的结构和基础设施是千差万别的。一个企业系统可能包含1960年延续过来的老式大型机，并且与现代系统配合使用。在过去的10年间，网络和因特网已经把老式系统与现代系统集成在一起了。

企业中之所以要包含老式系统是为了避免把这些系统上的核心业务处理转移到现代系统时花费大量的资金。但是由于需要继续维护老式系统、引入新系统以及把两者与来自其他机构的系统合并起来，因此企业系统变得非常复杂。

我们来考虑一家金融服务公司，它已经存在30年了。30年前，该公司把其核心业务处理定义在一台大型机上，使用COBOL语言完成。多年以后，该公司经过了多次收购与合并，后来加入的其他那些公司的业务处理也与原有的系统集成起来。这些业务都采用当前最先进的技术开发出了相关应用程序。结果是，现在该公司有了一个非常复杂的结构，把许多不同的硬件和软件系统链接在一起。当然，这些系统不仅拥有内部链接，还有通向Web的外部链接。

Web所在的分布式环境可供职员们在自己办公室里的一台PC机前工作，与互连的任意其他系统或者资源交互，既可以在业务范围内，也可以在业务范围之外。外部系统（例如供货商的系统）也包含在网络中，并且以此与企业员工和系统通过Web通信。这就提出一个问题，只有建立一种公共语言才能使这些系统相互通信。

网络 and 协议

在开发分布式计算系统的时候，基层的物理连接是建立这些系统的基础。如果去掉这些网络，分布式应用程序将失去其大部分的使用价值。基本的网络结构对开发者经常是透明的。Java语言的“write once, run anywhere（只编写一次，可以在任何地方应用）”这种思想提供一种强大的网络API工具，使得开发分布式应用程序更容易，也使我们能够了解进行开发的网络拓扑。当我们开始在分布式环境中处理servlet的时候，应该首先简单观察一下计算机之间的物理连接以及系统链路。

在一个网络系统中，所有彼此链接的系统都是互连的，要么通过有线网络，要么通过无线网络。网络就是计算机系统及其上运行的软件之间的一个通信链路。在图1.1中显示了三种基本的网络拓扑：

- 星型（star）拓扑用于把多台计算机连接到一个中心点，这个中心点经常称为集线器（hub）
- 环型（ring）拓扑用于把多台计算机连接成一个封闭的环形，在这里每一台计算机都与接下来的计算机连接起来，直到形成一个封闭的环
- 总线型（bus）拓扑则把多台计算机连接到一条共享的介质上，整个系统通过这条线路彼此通信

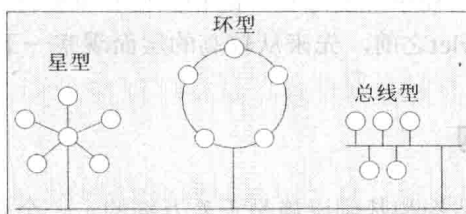


图1.1

当然，这些并不是在网络中可以实现的拓扑结构的全部。上述拓扑主要用于局域网（LAN）中。它们是最普通的结构，大多数其他备选的拓扑则是这几种型式的衍生或者结合。随着新的网络技术和通信技术的发展，如蓝牙（Bluetooth）技术，企业网络的拓扑还会变得更加复杂。要求网络（以及网络管理员）支持日新月异的技术也是一个持续不断的挑战。

网络被设计成一种通信渠道，可供不同的以及在其他方面不兼容的系统能够连接到相同的网络并且进行通信。我们都知道自己的网络怎样在相同的网络中包含不同的操作系统和不同的设备。我们可以使用一个浏览器来与一台服务器通信，而该服务器可能运行在完全不同的另外一种操作系统上。因此，问题是不兼容的系统如何彼此通信？