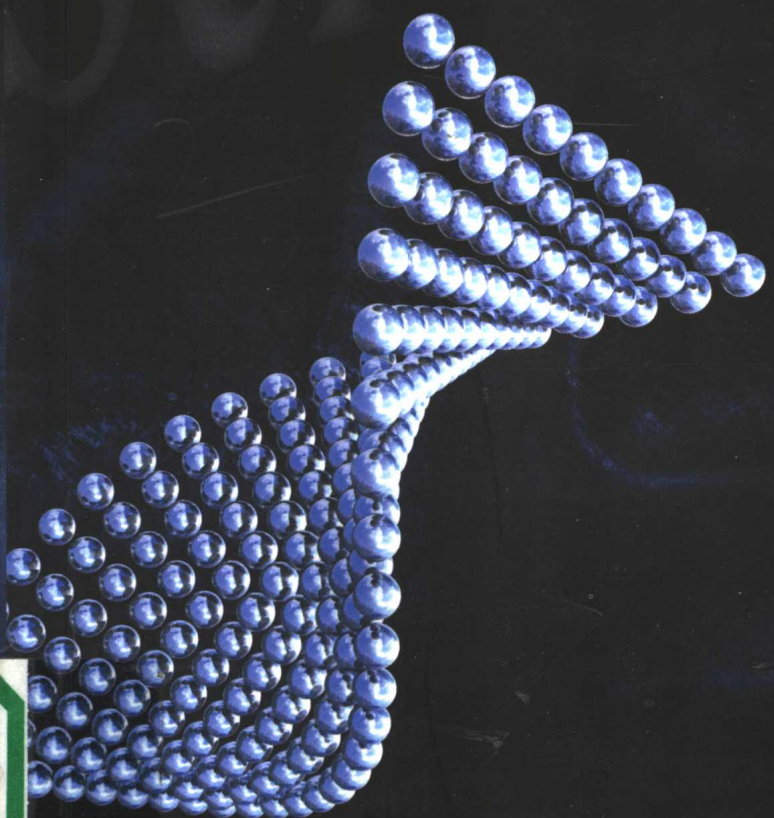
Servlet/JSP
程序设计技术与实例

邹华　方卫宁　邹蓉　编著



最新的技术：基于Servlet 2.2/2.3，JSP 1.1/1.2

全面的介绍：涵盖程序设计基础及高级应用技术

大量的实例：提供在电子商务和流行网站方面的综合实例

人民邮电出版社
www.pptph.com.cn

Servlet/JSP
程序设计技术与实例

邹 华 方卫宁 邹 蓉 编著

人民邮电出版社

图书在版编目（CIP）数据

Servlet/JSP 程序设计技术与实例/邹华，方卫宁，邹蓉编著. —北京：人民邮电出版社，2001.10

ISBN 7-115-09711-9

I. S… II. ①邹… ②方 .. ③邹… III. ①JAVA 语言—程序设计
②程序语言，JSP—主页制作 IV. TP312

中国版本图书馆 CIP 数据核字（2001）第 069245 号

内 容 提 要

Servlet/JSP 是开发 Java 服务器端应用程序的重要技术。目前 EJB+Servlet+JSP 几乎成为电子商务的开发标准。本书全面介绍了与编写 Servlet/JSP 服务程序相关的各种技术。主要内容包括 Servlet 的生命周期、接受请求、生成响应、会话管理、JSP 的语法综述、使用 JSP 的标记扩展机制、Servlet/JSP 的环境、利用 JDBC 访问数据库、Applet 和 Servlet/JSP 服务程序通信的各种技术、Servlet/JSP 的安全，以及在 Java 网络程序设计中服务器侧应有的体系结构。书中列举了大量应用实例，使读者能够全面掌握使用 Servlet/JSP 编写 Java 服务器端网络应用程序的各种技术和技巧。

本书适用所有对 Java，特别是开发 Java 服务器端应用程序感兴趣的读者使用和参考。

Servlet/JSP 程序设计技术与实例

◆ 编　　著　邹　华　方卫宁　邹　蓉
责任编辑　邹文波
执行编辑　苗　颖

◆ 人民邮电出版社出版发行　北京市崇文区夕照寺街 14 号
邮编　100061　　电子函件　315@ pptph com cn
网址　http://www pptph com cn
读者热线·010-67129212　010-67129211(传真)
北京汉魂图文设计有限公司制作
北京鸿佳印刷厂印刷
新华书店总店北京发行所经销

◆ 开本：787 × 1092　1/16
印张：24 75
字数：596 千字　　2001 年 10 月第 1 版
印数：1 – 5 000 册　　2001 年 10 月北京第 1 次印刷

ISBN 7 – 115 – 09711 – 9/TP·2509

定价：38 00 元

前　言

在当前的信息时代，Internet 起着主导作用。作为计算机网络的主干，Internet 的主要工作之一便是提供信息共享的平台、实现基于 Web 的访问。

传统的 Web 服务器只提供静态网页的服务，并不能产生动态网页。新的挑战在于创建交互的基于 Web 的应用程序。在这些程序中，页面的内容是基于用户的请求或者系统的状态，而不是预先定义的文字。

开发 Web 应用程序的早期方案是使用 CGI。随后，一些 Web 服务器供应商通过为他们的服务器提供“插件”和 API 来简化 Web 应用程序的开发。例如，ASP 技术使得在 Web 页面上创建动态内容更加容易，但是也只能工作在微软的产品上。Servlet 技术是解决这一问题的一种较新的方案。它使得用 Java 语言编写交互式服务器端应用程序变得容易，然而 Servlet 技术仍存在一定的不足。采用这种方法生成带有动态内容的页面仍然需要应用程序的开发技巧。JSP 技术以 Servlet 技术为基础，并在许多方面作了改进，为创建显示动态生成内容的 Web 页面提供了一个更简捷快速的方法。它的设计目的是使得构造基于 Web 的应用程序更加容易和快捷，而这些应用程序能够与各种 Web 服务器、应用服务器、浏览器和开发工具共同工作。

Servlet 和 JSP 各有适用的场合。Servlet 是完全的 Java 程序，适用于处理逻辑多的场合；而 JSP 融合了 HTML 的内容，适用偏重于显示的场合。它们可以互相协作，互相补充对方的不足。当被整合入 J2EE 后，Servlet/JSP 技术提供了企业级的扩展性和性能，这对于在虚拟企业中部署基于 Web 的应用是必需的。

全书共分 16 章。第 1 章至第 10 章，介绍了 Servlet/JSP 程序设计技术的基础；第 11 章至第 14 章，分别介绍了在 Web 应用程序中的高级应用开发；第 15 章和第 16 章，主要介绍了两个综合应用实例：网上购书系统和班级网站。

事实上，除了第三部分实战篇的综合实例外，其他部分中也有大量的实例程序。本书的主要意图就是要通过大量的例子使读者能够尽快地掌握 Servlet/JSP 编程技术和开发技巧，成为 Servlet/JSP 高手。

本书的前言、第 1 章至第 8 章由邹华编写，第 9 章、第 10 章、第 12 章、第 13 章及第 14 章由方卫宁编写，第 11 章、第 15 章和第 16 章由邹蓉编写。

由于时间和水平有限，书中难免有不足之处，敬请广大读者不吝赐教。

由于篇幅有限，书中实例的程序代码未完全列出，读者可以从 http://www.ucbook.com 的“资料下载”中下载程序代码。

编　者

目　录

第 1 章　Servlet/JSP 概论

1.1　客户机/服务器结构

客户机/服务器（Client/Server，C/S）结构是近几年非常受欢迎的一种分布式计算模式，其优势在于广泛地采用了网络技术，将系统中的各部分任务分配给分布在网络上的担任不同角色的计算机。它把较复杂的计算和管理任务交给网络上的高档机——服务器（Server），而把一些频繁与用户打交道的任务交给前端较简单的计算机——客户机（Client），从而实现了网络上信息资源的共享。

客户机/服务器结构按照应用逻辑在客户端和服务器端的分布情况，通常被划分为两层或多层体系结构。在最简化的情况下，客户机/服务器体系必须具有一个客户层和一个服务器层。客户机/服务器的两层体系结构如图 1-1 所示。它由两部分组成。前端是客户机，为每个用户所专有，与用户直接进行信息交互，如管理用户接口、数据处理和报告请求等。后端是服务器，具有由多个用户共享的信息与功能，如管理共享外设、控制对共享数据库的操作、接受并应答客户机的请求等。这种体系结构将一个应用系统分成两大部分，由多台计算机分别执行，使它们有机地结合在一起，协同完成整个系统的应用，从而达到系统中软、硬件资源最大限度的利用。

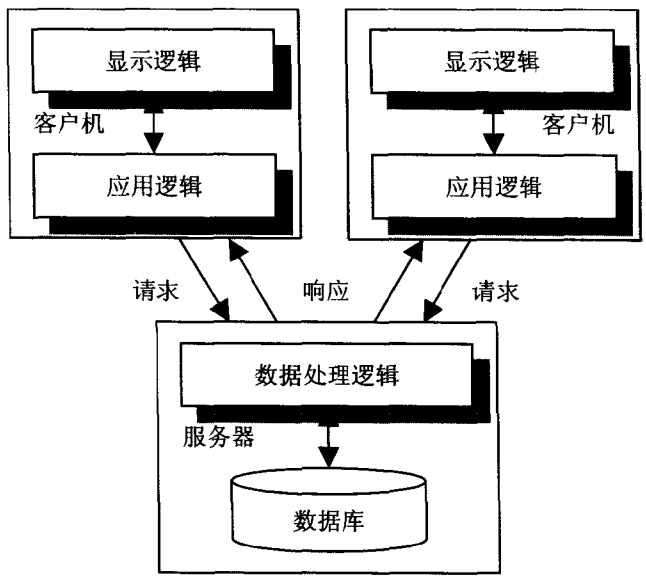


图 1-1　两层结构的客户机/服务器模式

尽管两层结构的客户机/服务器模式有很多优点，而且在实际应用中也发挥了重要的作用，但是它还是存在着不足，其中最主要的一个是维持客户端所需的开销很大。由于每个客户端都

包含了独立于服务器的应用逻辑，所以位于每个客户端的应用逻辑都必须单独维护。此外，客户端可运行于不同的平台上，由此产生了基于不同平台的、版本各异的应用。

实际上，任何一个应用系统，从简单的单机系统到复杂的网络计算，都由三部分组成：显示逻辑部分（表示层）、事务处理逻辑部分（功能层）和数据处理逻辑部分（数据层）。表示层的功能是实现与用户的交互；功能层的功能是进行具体的运算和数据的处理，数据层的功能是实现对数据库中的数据进行查询、修改和更新等任务。这样就自然导致了三层或多层的客户机/服务器结构的出现。

在三层客户机/服务器计算模式中，位于客户端和服务器之间的是中间层（Middle Tier）。这个中间层由应用服务器组成。它包含了大量的基于三层模式的应用逻辑，即功能层的事务处理部分，而前端的客户机负责表示层的显示逻辑部分，后端的数据服务器负责数据层的数据处理逻辑部分。在此模式中客户端为“瘦客户端（Thin Client）”，应用逻辑位于单独一层，因而在每个节点上很容易维护。而且中间层体系结构的设计能优化服务器性能，因为它分担了服务器的一些功能和负荷，能使服务器实现负载均衡。三层客户机/服务器计算模式如图 1-2 所示。

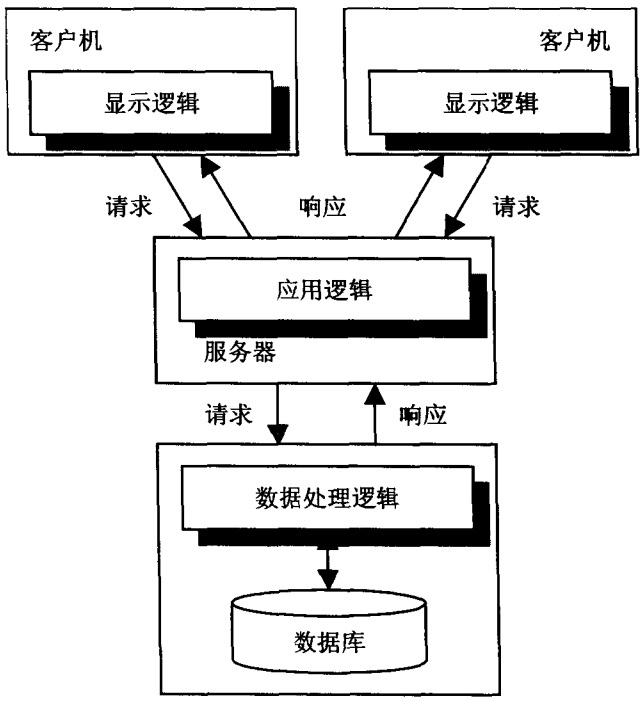


图 1-2　三层结构的客户机/服务器模式

1.2　浏览器/服务器结构

随着 Internet 越来越广泛的应用，一种新兴的体系结构浏览器/服务器（Browser/Server）应运而生，并获得飞速发展，成为众多厂家争相采用的新型体系结构。本质上，Browser/Server 也是一种 Client/Server 结构，它是一种由传统的二层 Client/Server 结构发展而来的三层 Client/Server 结构在 Web 上应用的特例。

在 Browser/Server 三层体系结构下，表示层（Presentation）、功能层（Business Logic）、数据层（Data Service）被割成 3 个相对独立的单元。

- 表示层——Web 浏览器。在表示层中包含系统的显示逻辑，位于客户端。它的任务是由 Web 浏览器向网络上的某一 Web 服务器提出服务请求，Web 服务器对用户身份进行验证后用 HTTP 协议把所需的主页传送给客户端，客户机接受传来的主页文件，并把它显示在 Web 浏览器上。
- 功能层——具有应用程序扩展功能的 Web 服务器。在功能层中包含系统的事务处理逻辑，位于 Web 服务器端。它的任务是接受用户的请求，进行相应的处理，如查询数据库，而后将数据处理的结果返回给 Web 服务器，再由 Web 服务器传送回客户端。
- 数据层——数据库服务器。在数据层中包含系统的数据处理逻辑，位于数据库服务器端。它的任务是接受 Web 服务器对数据库操作的请求，实现对数据库查询、修改和更新等功能，把运行结果返回给 Web 服务器。

实际上，三层的 Browser/Server 体系结构是把两层 Client/Server 结构的事务处理逻辑模块从客户机的任务中分离出来，由单独组成的一层来负担其任务，这样客户机的压力大大减轻了，把负荷均衡地分配给了 Web 服务器，于是由原来的两层的 Client/Server 结构转变成三层的 Browser/Server 结构。这种三层体系结构如图 1-3 所示。

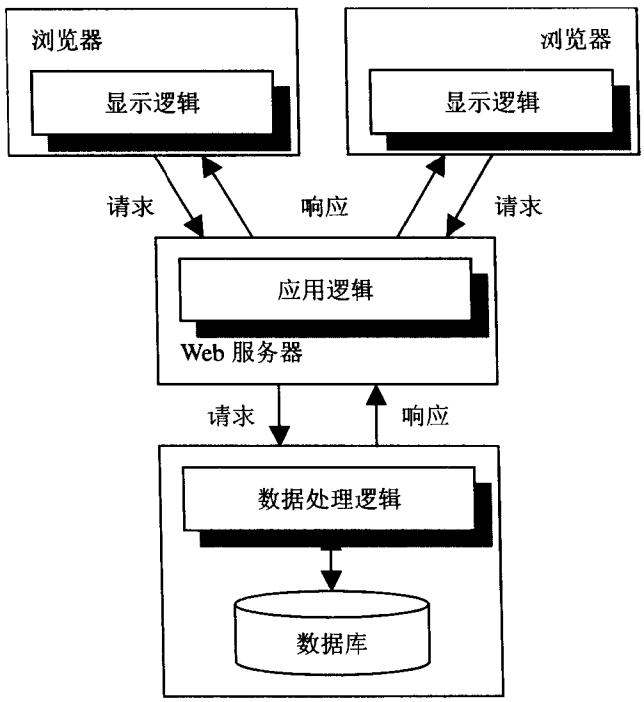


图 1-3　三层 Browser/Server 结构

Browser/Server 体系结构与 Client/Server 体系结构相比不仅具有 Client/Server 体系结构的优点，而且又有 Client/Server 体系结构所不具备的独特优势。

- 开放的标准。Client/Server 所采用的通信协议往往是专用的。Browser/Server 所采用的标准，如 HTTP、HTML 等，都是开放的、非专用的，是经过标准化组织所确定的，保证了其

应用的通用性和跨平台性。

- 较低的开发和维护成本。Client/Server 的应用必须开发出专用的客户端软件，无论是安装、配置还是升级都需要在所有的客户机上实施，极大地浪费了人力和物力。Browser/Server 的应用只需在客户端装有通用的浏览器即可，维护和升级工作都在服务器端进行，不需对客户端进行任何改变，故而大大降低了开发和维护的成本。
- 使用简单，界面友好。Client/Server 用户的界面是由客户端软件所决定的，其使用的方法和界面各不相同，每推广一个 Client/Server 系统都要求用户从头学起，难以使用。Browser/Server 用户的界面都统一在浏览器上，浏览器易于使用、界面友好，不须再学习使用其他的软件，一劳永逸地解决了用户的使用问题。
- 客户端简单。Client/Server 的客户端具有显示与处理数据的功能，对客户端的要求很高，是一个“胖”客户机。Browser/Server 的客户端不再负责数据库的存取和复杂数据计算等任务，只需要根据结果数据中指明的格式对其进行显示，充分发挥了服务器的强大作用，这样就大大地降低了对客户端的要求，客户端变得非常“瘦”。
- 系统灵活。Client/Server 系统的三部分模块中有一部分需改变就要关联到其他模块的变动，使系统极难升级。Browser/Server 系统的三部分模块各自相对独立，其中一部分模块改变时其他模块不受影响，系统改进变得非常容易，且可以用不同厂家的产品来组成性能更佳的系统。
- 保障系统的安全性。在 Client/Server 系统中由于客户机直接与数据库服务器进行连接，用户可以很轻易地改变服务器上的数据，无法保证系统的安全性。Browser/Server 系统在客户机与数据库服务器之间增加了一层 Web 服务器，使两者不再直接相连，客户机无法直接对数据库操纵，有效地防止了恶意用户的非法入侵。

在浏览器/服务器三层结构中，位于中间层的 Web 服务器处于非常重要的地位。它不但负责系统的事务处理逻辑以生成结果数据，而且要负责结果数据的显示格式。它的核心就是具有很强的动态数据的生成和发布能力。目前，支持动态 Web 的技术主要有 CGI、ASP、PHP，另外，就是本书所要介绍的 Servlet 和 JSP 了。从本章的后续介绍中，我们将会看到 Servlet/JSP 技术的特点以及优势。

1.3 Servlet 概述

1.3.1 什么是 Servlet

Servlet 译为服务器小程序，运行于服务器端，与运行于客户端的小应用程序 Applet 相对应。它们都是由 Java 编写的小型程序，都没有 main 方法，只有一些特定的方法用于启动、执行和退出，但略有不同的是 Servlet 并不提供用户界面。

Servlet 用于扩充 Java-enabled 服务器功能，目前常用于增加 Web 服务器的功能。传统的 Web 服务器只提供静态网页的服务，并不能产生动态网页。静态 HTML 网页对于显示相对静态的内容是不错的选择。新的挑战在于创建交互的基于 Web 的应用程序。在这些程序中，页面的内容是基于用户的请求或者系统的状态，而不是预先定义的文字。为了解决这一问题，我们通常会另外编写程序，用于产生动态网页或是扩充 Web 服务器的基本功能，这种程序就称

为 Web 应用程序。它可以非常简单，也可以非常复杂。

开发 Web 应用程序的早期方案是开发人员使用公共网关接口（Common Gateway Interface，CGI）编写与接口相关的单独的程序，以及基于 Web 的应用程序。后者通过 Web 服务器来调用前者。这个方案有着严重的扩展性问题——每一个对 CGI 的请求都将导致在服务器上新增一个进程。如果多个用户并发地访问该程序，这些进程将消耗该 Web 服务器所有的可用资源，并且系统性能降低到极其低下的地步。

此外，一些 Web 服务器供应商已尝试通过为他们的服务器提供“插件”和 API 来简化 Web 应用程序的开发。这些解决方案是与特定的 Web 服务器相关的，不能解决跨多个供应商的解决方案的问题。例如，微软的 Active Server Pages（ASP）技术使得在 Web 页面上创建动态内容更加容易，但是也只能工作在微软的 IIS 和 Personal Web Server 上。

Servlet 是解决这一问题的一种较新的技术。它使得用 Java 语言编写交互式服务器端应用程序变得容易。实际上，早在 1996 年，大家就已发现利用 Java 语言本身具有的优点来开发服务器端的应用程式是非常有利的，所以有软件厂商开始发展一些技术以使服务器端的程序开发更具效率。例如 Netscape 的“server-side applets”、World Wide Web Consortium 的“resource”和 O'Reilly 的“servlets”（这里指的是 O'Reilly 自行开发的技术，也称 Servlets），但这些技术都必须搭配着特殊的 Server，或者只专为某些问题所设计。接着在 1997 年，JavaSoft（Sun Microsystems）提出了 Java Servlet。它结合上述各家的技术成为一个单一、标准、通用的机制。Servlet 可以在 Java-based 或 non-Java-based 的 server 上执行，目前 Apache、IIS、Java Web Server、Jigsaw Web Server、Novell NetWare、Personal Web Server 及 Zeus Web Server 等 Web 服务器都已支持 Servlet。

Servlet 技术源自于请求/响应（Request/Response）模式。例如在 Web 应用中，Servlet 可用于接收来自 Web 浏览器的 HTTP 请求，动态地生成响应（可能要查询数据库来完成这项请求），然后发送包含处理结果的 HTTP 响应到浏览器，如图 1-4 所示。

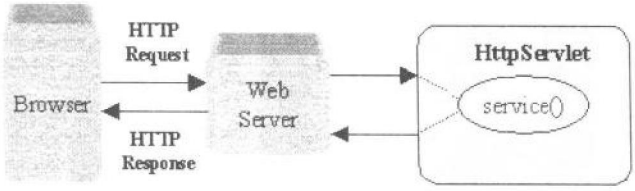


图 1-4　Servlet 请求/响应模式

需要指出的是，尽管目前 Servlet 常用于 Web 服务器，但由于 Servlet 的 API 并不是专为 HTTP 协议而设计的，它没有预设任何服务器的执行环境或是协议，可以依使用者的要求设计自己的协议，所以 Servlet 并不要求设计者将其固定在特定种类的服务器环境下，即它可以被用来扩充任何种类的服务器，如 FTP 服务器。在 FTP 服务器中，每个 ftp 指令都可对应一个独立的 Servlet，当有新的指令规格产生时，只要加入新的 Servlet 即可。

1.3.2　Servlet API

SUN 公司为 Servlet 的开发提供了一个标准的应用程序接口（*Application Programming* Interface，API），保证了 Servlet 开发的一致性。该 API 是与协议无关的，没有限定其在网络