

Ajax+JSP

网站开发 从入门到精通

梁文新 宋强 王占中 等编著



附
光
盘

- 16章完整实例源代码
- 18段全程配音教学视频
- 6个综合实例配置文件和代码
- Tomcat、Eclipse和JDK 6.0安装文件



清华大学出版社

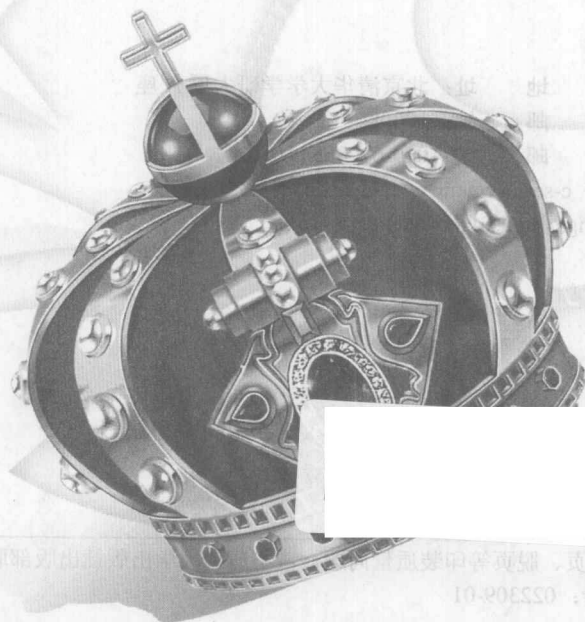
TP393.092/1052D

2008

Ajax+JSP

网站开发 从入门到精通

梁文新 宋强 王占中 等编著



清华大学出版社
北京

内 容 简 介

本书通过大量的实例全面深入地阐述了 Ajax+JSP 编程技术。全书共分为 16 章, 包括 JSP 的基础知识、JSP 访问外部数据、JSP 高级技术、Ajax 概述、Ajax 相关技术、Ajax 开发基础、Ajax 开发模式、Ajax 框架、Ajax 常用技巧。本书最后介绍了 6 个开发实例, 包括聊天室程序、留言本程序、通讯录程序、购物车程序、相册程序和 Blog 程序。配套光盘提供了本书源代码(包括 6 个完整的应用程序)和 Ajax+JSP 软件配置以及使用教学视频。

本书适合于中、高级 Ajax+JSP 网站开发人员, 特别适合于有编程基础, 希望全面学习 Ajax+JSP 技术, 想提高实际应用能力的读者群体。

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

Ajax+JSP 网站开发从入门到精通 / 梁文新, 宋强, 王占中等编著. —北京: 清华大学出版社, 2008.7
ISBN 978-7-302-17098-3

I. A… II. ①梁… ②宋… ③王… III. ①计算机网络—程序设计 ②JAVA 语言—主页制作—程序设计 IV. TP393.09

中国版本图书馆 CIP 数据核字(2008)第 022591 号

责任编辑: 夏兆彦

责任校对: 徐俊伟

责任印制: 杨 艳

出版发行: 清华大学出版社

地 址: 北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编: 100084

社总机: 010-62770175

邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者: 北京市清华园胶印厂

装 订 者: 三河市兴旺装订有限公司

经 销: 全国新华书店

开 本: 203×260 印 张: 30.75

字 数: 803 千字

附光盘一张

版 次: 2008 年 7 月第 1 版

印 次: 2008 年 7 月第 1 次印刷

印 数: 1~4000

定 价: 59.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题, 请与清华大学出版社出版部联系调换。
联系电话: 010-62770177 转 3103 产品编号: 022309-01

FOREWORD

前言

在目前的 Web 2.0 热潮中, Ajax 已成为人们谈论最多的技术术语。Ajax (Asynchronous JavaScript and XML) 是多种技术的综合, Ajax 技术基于被各大浏览器和平台都支持的公开标准的技术, 组成 Ajax 技术的大多数技术都经过很多年的实践考验。它使用 XHTML 和 CSS 标准化呈现, 使用 DOM 实现动态显示和交互, 使用 XML 和 XST 进行数据交换与处理, 使用 XMLHttpRequest 对象进行异步数据读取, 使用 JavaScript 绑定和处理所有数据, 更重要的是, 它打破了使用页面重载的惯例技术组合, 可以说, Ajax 已成为 Web 开发的重要工具。

Google Earth 的广泛应用吸引了 Web 开发人员, 由此揭开了罩在 Ajax 上的面纱。Ajax 技术的广泛采用, 已被业内人士证明了其受市场的欢迎程度和该技术组的正确。该技术的使用者都成为了胜利者, 包括 Google、Yahoo、Amazon 和 Microsoft 等。

本书通过大量的实例全面而深入地阐述了 Ajax 编程技术的方方面面。全书共分为 16 章, 其内容如下所述。

第 1 章为 JSP 概述, 全面地介绍了 JSP 产生和发展的过程, 并重点论述了 JSP 环境的构建与配置。

第 2 章为 JSP 的基础知识, 主要介绍了两部分内容, 即 JSP 页面组成元素和 JSP 内置对象。前部分从页面声明开始介绍, 然后介绍程序片、表达式、指令标签和动作标签等; 后部分从 JSP 内置对象着手, 详细地介绍了常用的内置对象, 如 request、session 等。

第 3 章为 JSP 访问外部数据, 主要介绍了利用 JSP 技术获取外部数据。JSP 技术以流的形式获取文件中数据, 常见的流有字节流和字符流。本章地重点在于 JSP 获取数据库中的数据, 以实例的方式介绍了操作数据库中的数据。

第 4 章为 JSP 的高级技术, 主要阐述了两种非常流行的 JSP 技术, 即 JavaBean 和 Servlet。本章前部分介绍了利用 JSP 页面调用 JavaBean、JavaBean 的作用域范围、JSP 页面与 JavaBean 之间传递数据; 本章后部分主要介绍 Servlet 技术, 从简单的 Servlet 类入手, 介绍了创建 Servlet 类的步骤和配置、Servlet 技术常用接口。在本章的后面给出了一个结合 JSP、JavaBean 和 Servlet 三种技术的 MVC 程序。

第 5 章为 Ajax 概述, 主要介绍了 Ajax 技术的相关概念。本章从 Web 2.0 的形成和发展过程开始讲起, 论述了 Ajax 的概念和实现的工作原理。

第 6 章为 Ajax 相关技术简介, 主要介绍了 Ajax 的相关技术, 如 XHTML、CSS、XML、DOM、XMLHttpRequest 和 JavaScript 技术, 本章是 Ajax 技术的基础章节。

第7章为 Ajax 的开发基础，主要介绍了 Ajax 客户端对发送和获取信息的信息格式处理。本章从传输数据的格式入手，介绍了常用的两种传送格式，即普通字符串和 XML 文件格式字符串。

第8章为 Ajax 开发模式，主要介绍了 Ajax 比较流行的开发模式，如 Façade、Observer、MVC 等模式。

第9章为 Ajax 框架，重点介绍了 Ajax 的流程框架，如 Dojo、MooTools 等。本章从 Ajax 的框架种类开始介绍，重点介绍基于浏览器的框架。

第10章为 Ajax 的常用技巧，主要介绍了 Ajax 技术在实际开发中，需要使用的 Ajax 技巧，如表单验证、自动刷新页面、级联菜单和数据库分页显示等。

第11章为聊天室程序，主要介绍了使用 Ajax+JSP 技术实现聊天室。该实例以记事本为开发工具，Tomcat 为服务器，以简陋的条件实现了 Ajax 技术，更能体现 Ajax 的本质，这是本书的一大特点。

第12章为留言板程序，主要介绍了 Ajax 实现留言板的实例。本章的留言板程序具有添加留言、删除留言和查看留言等功能。

第13章为通讯录程序，主要介绍了同学通讯录的实现过程，具有添加、删除、修改等功能。

第14章为购物车实例程序，主要介绍了购物车实现过程，具有查看购物车、添加购物车等功能。

第15章为相册系统程序，主要介绍了 Ajax 实现网上浏览相片，上传相片的过程。本章是以 Eclipse 为开发工具实现的。

第16章为 Blog 系统程序，主要介绍了现在比较流行的 Blog 的创建过程。实现文章管理、类别管理、用户管理等功能。该程序的实现过程中，包含了对数据库各种不同的操作，如分页、更新、删除和采用数据库函数等。

本书全面介绍了 Ajax+JSP 面向实践的开发和应用知识，具有下面的特点。

- **内容全面** 本书是一本大全性质的 Ajax+JSP 编程图书，突出介绍了 Ajax+JSP 面向实际的开发知识。读者学习本书之后，可以全面地掌握 Ajax+JSP 动态网站开发的实践知识。
- **实例丰富** 全书每个知识点的讲解都配有大量可实际运行的实例，读者可以边学习边实践，快速、全面地掌握 Ajax+JSP 的开发方法和技巧。书中最后提供了 6 个典型开发实例，覆盖了 Ajax+JSP 开发和部署网站的全部过程。

配套光盘提供了本书源代码（包括 6 个完整的应用程序）和 Ajax+JSP 软件配置与使用教学视频。读者只要按照书中的范例上机练习、举一反三，就可以根据自己的需要开发出功能强大的 Web 动态网站。

本书适合于中、高级 Ajax+JSP 网站开发人员，特别适合于有编程基础，希望全面学习 Ajax+JSP 技术，提高实际应用能力的读者群体。

除了封面署名人员之外，参与本书编写的人员还有李乃文、张仕禹、夏小军、赵振江、李振山、李文采、吴越胜、李海庆、何永国、李海峰、陶丽、吴俊海、安征、张巍屹、崔群法、王咏梅、康显丽、辛爱军、牛小平、贾栓稳、王立新、苏静、赵元庆、郭磊、徐铭、李大庆、王蕾、张勇、郝安林等。在编写过程中难免会有疏漏之处，希望大家谅解，并欢迎读者与我们联系，帮助我们改正提高。

CONTENTS

目 录

第 1 章 JSP 概述	1
1.1 JSP 简介	1
1.1.1 JSP 技术概述	1
1.1.2 JSP 的开发模式	4
1.2 JSP 的开发环境	6
1.2.1 安装和配置 JDK	6
1.2.2 安装和配置 Tomcat	9
1.2.3 JSP 的运行原理	13
1.3 JSP 开发实例	16
第 2 章 JSP 的基础知识	19
2.1 JSP 页面元素	19
2.2 JSP 脚本元素	21
2.2.1 声明	21
2.2.2 Java 脚本程序	24
2.2.3 表达式和注释	24
2.3 JSP 指令元素	25
2.3.1 页面设置指令 page	25
2.3.2 页面包含指令 include	27
2.4 JSP 动作元素	28
2.4.1 <jsp:include>文件包含	28
2.4.2 <jsp:forward>文件转向	30
2.4.3 其他动作指令	31
2.5 request 对象	32
2.5.1 request 简介	33
2.5.2 获取客户端基本信息	34
2.5.3 获取参数值	36
2.6 response 对象	38
2.6.1 response 简介	38
2.6.2 设置响应信息	39
2.6.3 页面重定向	40
2.7 session 对象	42
2.7.1 session 简介	42
2.7.2 绑定会话	43
2.8 application 对象	45
2.8.1 application 简介	45

2.8.2 绑定全局变量	46	4.3.2 追加数据 JavaBean	105
2.9 其他对象	47	4.4 Servlet 概述	107
2.9.1 out 对象	47	4.4.1 Servlet 简介	107
2.9.2 exception 对象	48	4.4.2 Servlet 的生命周期	109
2.9.3 page 和 config 对象	50	4.4.3 开发、部署一个简单的 Servlet	110
2.10 留言板实例	51	4.5 实现 Servlet	113
第 3 章 JSP 访问外部数据	55	4.5.1 Servlet 体系结构	113
3.1 File 类	55	4.5.2 创建 Servlet 类	115
3.2 流操作	59	4.6 Servlet 处理请求和响应	118
3.2.1 流简介	60	4.7 Servlet 中会话跟踪	121
3.2.2 使用字节流读写文件	60	4.8 JSP+Servlet+JavaBean 实例	123
3.2.3 使用字符流读写文件	63	第 5 章 Ajax 概述	127
3.3 文件操作	66	5.1 Web 2.0 简介	127
3.3.1 文件上传	66	5.1.1 Web 发展历史	127
3.3.2 文件下载	68	5.1.2 Web 2.0 简介	129
3.4 JDBC 简介	69	5.2 Ajax 结构及其意义	130
3.4.1 JDBC 的概念	69	5.2.1 传统 Web 应用解决方案	131
3.4.2 JDBC API 介绍	71	5.2.2 Ajax 技术概念	131
3.5 数据库操作	74	5.2.3 Ajax 相关技术简介	132
3.5.1 添加记录	74	5.2.4 Ajax 解决方案的优势	134
3.5.2 修改记录	76	5.3 深入理解 Ajax	135
3.5.3 删除记录	78	5.3.1 Ajax 的工作原理	135
3.5.4 查询记录	80	5.3.2 Ajax 的基本特征	136
3.5.5 分页显示记录	82	5.4 Ajax 异步请求实例	139
3.6 获取数据库信息	85	第 6 章 Ajax 相关技术简介	142
3.7 连接数据库的其他方式	87	6.1 XHTML	142
第 4 章 JSP 高级技术	90	6.1.1 XHTML 概述	142
4.1 JavaBean 简介	90	6.1.2 XHTML 与 HTML 的区别	145
4.1.1 JavaBean 介绍	90	6.1.3 XHTML 的语法	147
4.1.2 定义 JavaBean	92	6.1.4 XHTML 2.0 简述	157
4.2 使用 JavaBean	93	6.2 CSS	158
4.2.1 调用 JavaBean	93	6.2.1 CSS 语法	159
4.2.2 scope 作用域范围	96	6.2.2 引入样式表	162
4.2.3 setProperty 动作标记	99	6.2.3 CSS 属性	164
4.2.4 getProperty 动作标记	102	6.3 XML	170
4.3 开发 JavaBean 实例	103	6.3.1 XML 文档	170
4.3.1 获取记录集 JavaBean	103	6.3.2 XML 命名空间	173

6.3.3 文档类型定义 (DTD)	175
6.4 JavaScript	179
6.4.1 JavaScript 简介	179
6.4.2 JavaScript 的基本结构	181
6.4.3 表达式与运算符	182
6.4.4 流程控制	183
6.4.5 函数	185
6.4.6 事件驱动及事件处理	186
6.4.7 创建对象	188
6.4.8 JavaScript 中的对象	190
6.5 DOM	193
6.5.1 DOM 对象	193
6.5.2 使用 DOM 操作 HTML 文档	196
6.5.3 使用 DOM 解析 XML 文档	198
6.6 XMLHttpRequest	201
6.6.1 XMLHttpRequest 简介	201
6.6.2 XMLHttpRequest 的属性和方法	203
6.6.3 XMLHttpRequest 与服务器通信	205
6.6.4 XMLHttpRequest 实例	207
第 7 章 Ajax 的开发基础	211
7.1 HTTP 请求中包含的参数	211
7.1.1 发送普通请求	211
7.1.2 发送 XML 格式请求	217
7.2 处理服务器响应	222
7.2.1 处理普通文本响应	222
7.2.2 处理 XML 格式响应	225
7.3 Ajax 实例	229
第 8 章 Ajax 的开发模式	232
8.1 Ajax 与设计模式	232
8.1.1 设计模式	232
8.1.2 Ajax 与重构	234
8.1.3 一个使用重构的简单实例	234
8.2 Ajax 中常用的设计模式	237
8.2.1 Facade 模式	237
8.2.2 Observer 模式	238
8.2.3 Singleton 模式	239
8.3 MVC 模式	241

8.3.1 MVC 简介	241
8.3.2 Ajax 中的 MVC	242
第 9 章 Ajax 框架	249
9.1 Ajax 概述	249
9.2 Dojo 框架	252
9.2.1 Dojo 框架简介	252
9.2.2 安装 Dojo 框架	253
9.2.3 Dojo 标记介绍	255
9.2.4 创建静态页面	257
9.2.5 Dojo 事件处理	259
9.2.6 Dojo 异步调用	262
9.3 DWR 框架	263
9.3.1 DWR 框架概述	263
9.3.2 部署 DWR 框架	265
9.3.3 DWR 框架函数	267
9.3.4 身份验证	269
9.3.5 包含文件	271
9.4 MooTools 框架	273
9.4.1 MooTools 框架介绍	273
9.4.2 安装 MooTools 框架	274
9.4.3 MooTools 常用类 (一)	275
9.4.4 MooTools 常用类 (二)	277
9.4.5 MooTools 事件处理	283
9.4.6 MooTools 异步调用	284
第 10 章 Ajax 的常用技巧	288
10.1 实现表单数据验证	288
10.2 实现 Web 页面中的级联菜单	296
10.3 实现自动刷新页面	299
10.4 实现数据库分页	302
10.5 实现进度条	306
10.6 实现 Web 页面局部动态更新	311
第 11 章 聊天室	316
11.1 开发前的准备	316
11.1.1 聊天室分析	317
11.1.2 系统概述	317
11.1.3 数据库设计	318

11.2 代码实现	320
11.2.1 公共类编写	320
11.2.2 用户注册	322
11.2.3 用户登录	327
11.2.4 聊天室	330
11.2.5 查看个人聊天记录	334
11.2.6 删除聊天记录	338
11.3 实例演示	341

第 12 章 留言板

12.1 需求分析和系统概述	344
12.1.1 需求分析	344
12.1.2 系统概述	345
12.1.3 数据库设计	346
12.2 客户端代码	347
12.2.1 客户端原理分析	347
12.2.2 主界面客户端实现	348
12.2.3 显示页面客户端实现	352
12.2.4 校验页面客户端实现	353
12.2.5 删除页面客户端实现	355
12.3 服务器端实现	357
12.3.1 封装数据访问类	357
12.3.2 添加处理页面	358
12.3.3 显示处理页面	359
12.3.4 删除处理页面	362
12.3.5 身份校验处理页面	363
12.4 运行留言板程序	363

第 13 章 通讯录

13.1 需求分析和系统概述	366
13.1.1 需求分析	366
13.1.2 系统概述	367
13.1.3 数据库设计	368
13.2 客户端页面实现	368
13.2.1 通讯录输入页面	369
13.2.2 通讯录删除页面	372
13.2.3 通讯录显示页面	374
13.2.4 通讯录修改页面	376

13.2.5 通讯录简单查询页面	380
13.2.6 通讯录高级查询页面	381
13.3 服务器端页面实现	383
13.3.1 输入处理页面	383
13.3.2 删除处理页面	385
13.3.3 显示处理页面	387
13.3.4 修改处理页面	389
13.3.5 简单查询处理页面	390
13.3.6 高级查询处理页面	391
13.4 运行通讯录程序	392

第 14 章 购物车实例

14.1 开发前的准备	396
14.1.1 需求分析	396
14.1.2 系统设计	397
14.1.3 数据库设计	398
14.2 核心代码	400
14.2.1 服务器端代码	400
14.2.2 客户端代码	408
14.3 实例演示	411
14.3.1 登录/注册模块	411
14.3.2 商品类别模块	417
14.3.3 订单模块	419

第 15 章 网络相册系统

15.1 需求分析	424
15.2 系统分析	426
15.3 数据库设计	426
15.4 系统公共类	428
15.4.1 数据库对象类	428
15.4.2 JavaBean 类	431
15.4.3 业务逻辑类	433
15.4.4 公用工具类	434
15.5 配置 web.xml	439
15.6 页面显示	441
15.6.1 源码解析	441
15.6.2 运行时界面截图	445

第 16 章 Blog 系统.....	448		
16.1 Blog 介绍.....	448	16.6.1 注册 Blog 会员	456
16.2 系统概述	450	16.6.2 添加 Blog 文章	462
16.3 数据库设计	451	16.6.3 分页浏览文章标题	465
16.4 通用模块设计	452	16.6.4 阅读 Blog 文章	469
16.5 首页实现	454	16.6.5 Blog 文章删除	474
16.6 系统实现	456	16.6.6 Blog 会员登录	478

JSP 概述

第1章



学习目标 | Objective

JSP (Java Server Pages) 是由 Sun 公司以 Java 语言为脚本语言开发出来的一种动态网页制作技术, 主要完成网页中服务器动态部分的编写。该技术是在 Servlet 技术的基础上形成的, 并继承了 Java 语言的多种优势, 如安全性、支持多线程、平台无关性等。与其他动态网页技术相比较, 如 ASP、PHP 等, 具有运行速度快、安全等特点。

本章将简要介绍 JSP 的产生到现状, 并重点介绍构建 JSP 的运行平台, 即 Tomcat 和 JDK 的安装及配置。在本章最后给出一个 JSP+JavaBean+Servlet 实例, 演示 JSP 各部分的运行情况。



内容摘要 | Abstract

- 了解 Web 技术的产生和发展。
- 了解 JSP 的产生和应用。
- 了解 JSP 的技术优势。
- 掌握常见 JSP 的设计模式。
- 掌握 JDK 下载、安装、配置。
- 掌握 Tomcat 下载、安装。
- 掌握编写 JSP 工作原理。
- 了解 JSP 页面包含元素。
- 了解 JSP 的开发流程。

1.1 JSP 简介

要熟练掌握 JSP 技术, 首先需要了解 JSP 的产生和发展历史, 明确 JSP 技术在 Web 技术中所占有位置, JSP 技术能够解决的问题。在本小节中, 将主要介绍 JSP 形成的必然原因, 与其他动态技术相比所具有的优势。

1.1.1 JSP 技术概述

在 Internet 发展的最初阶段, HTML 语言只能在浏览器中展现静态的文本或图像信息, 这无法满足人们对信息丰富性和多样性的强烈需求。随着 Internet 和 Web 技术应用到商业领域, Web

技术功能越来越强大。目前，解决 Web 动态网站的开发技术很多，如 Servlet，JSP，ASP，PHP 等，都得到了广泛应用。JSP 是它们中的佼佼者。

JSP (Java Server Pages) 是由 Sun Microsystem 公司于 1999 推出的新技术，是基于 Java Servlet 以及整个 Java 体系的 Web 开发技术。利用这一技术可以建立先进、安全和跨平台的动态网站。在传统的网页 HTML 文件 (*.htm, *.html) 中加入 Java 程序片段 (Scriptlet) 和 JSP 标记，就构成了 JSP 网页 (*.jsp)。Web 服务器在收到访问 JSP 网页的请求时，首先执行其中的程序片段，然后将执行结果以 HTML 格式返回给客户。程序片段可以操作数据库、重新定向网页、发送 E-mail 等等，这就是建立动态网站所需要的功能。JSP 所有程序操作都在服务器端执行，网络上传送给客户端仅是得到的结果，对客户浏览器的要求最低。

Servlet 是 JSP 技术的发展前身，它是 Java 技术对 CGI 编程的回应。Servlet 程序在服务器端运行，动态生成 Web 页面。Servlet 由 Java 语言编写，运行在 Tomcat 服务器中，能够主动生成 HTML 标记和客户端需要的数据，并能够将生成的数据返回到客户端。与传统的 CGI 和许多其他类似 CGI 的技术相比，Java Servlet 具有更高的效率，更容易使用，功能更强大，具有更好的可移植性，更节省投资。其详细信息如表 1-1 所示。

表 1-1 Servlet 技术优势

优势	说明
高效	在传统的 CGI 中，每个请求都要启动一个新的进程，如果 CGI 程序本身的执行时间较短，启动进程所需要的开销很可能反而超过实际执行时间。而在 Servlet 中，每个请求由一个轻量级的 Java 线程处理（而不是重量级的操作系统进程）； 在传统 CGI 中，如果有 N 个并发的对同一 CGI 程序的请求，则该 CGI 程序的代码在内存中重复装载了 N 次；而对于 Servlet，处理请求的是 N 个线程，只需要一份 Servlet 类代码。在性能优化方面，Servlet 也比 CGI 有着更多的选择，比如缓冲以前的计算结果，保持数据库连接的活动等
方便	Servlet 提供了大量的实用工具例程，例如自动地解析和解码 HTML 表单数据、读取和设置 HTTP 头、处理 Cookie、跟踪会话状态等
功能强大	在 Servlet 中，可以完成许多使用传统 CGI 程序很难完成的任务。例如，Servlet 能够直接和 Web 服务器交互，而普通的 CGI 程序不能。Servlet 还能够各个程序之间共享数据，很容易实现数据库连接池之类的功能
可移植性好	Servlet 用 Java 语言编写，Servlet API 具有完善的标准。因此，为 I-Planet Enterprise Server 写的 Servlet 无需任何实质上的改动即可移植到 Apache、Microsoft IIS 或者 WebStar。所有主流服务器都直接或间接通过插件支持 Servlet
节省投资	不仅有许多廉价甚至免费的 Web 服务器可供个人或小规模网站使用，而且对于现有的服务器，如果它不支持 Servlet 的话，要加上这部分功能也往往是免费的（或只需要极少的投资）

但遗憾的是，Servlet 具有一个致命缺点，就是所有响应代码都是通过 Servlet 程序生成的，如 HTML 标记。一个 Servlet 程序，其中大量的代码都是用来生成这些 HTML 标记响应代码，只有其中少部分代码用作数据的处理和响应。并且开发 Servlet 程序起点要求较高，Servlet 产生之后，没有像 PHP 和 ASP 那样，快速得到应用。故 Sun 公司在结合了 Servlet 技术和 ASP 技术等特点，又推出了 JSP 技术，JSP 技术完全继承了 Servlet 技术的优势，并具备了一些新的优势。JSP 相对于 PHP 和 ASP 技术有下面几种优势，如表 1-2 所示。

表 1-2 JSP 优势

优势	说明
数据内容和显示分离	使用 JSP 技术, Web 开发人员可以使用 HTML 或者 XML 标记来设计和格式化最终页面。使用 JSP 标记或者小脚本来产生页面上的动态内容。产生内容的逻辑被封装在标记和 JavaBeans 群组中, 并且捆绑在小脚本中, 所有的脚本在服务器端执行。如果核心逻辑被封装在标记和 Beans 中, 那么其他人, 如 Web 管理人员和页面设计者, 能够编辑和使用 JSP 页面, 而不影响内容的产生。在服务器端, JSP 引擎解释 JSP 标记, 产生所请求的内容(例如, 通过存取 JavaBeans 群组, 使用 JDBC 技术存取数据库), 并且将结果以 HTML(或者 XML)页面的形式发送回浏览器。这有助于作者保护自己的代码, 而又保证任何基于 HTML 的 Web 浏览器的完全可用性
可重用组件	绝大多数 JSP 页面依赖于可重用且跨平台的组件(如 JavaBeans 或者 Enterprise JavaBeans)来执行应用程序所要求的更为复杂的处理。开发人员能够共享和交换执行普通操作的组件, 或者使得这些组件为更多的使用者或者用户团体所使用。基于组件的方法加速了总体开发过程, 并且使得各种群组组织在他们现有的技能和优化结果的开发努力中得到平衡
采用标记简化页面开发	Web 页面开发人员不会都是熟悉脚本语言的程序设计人员。JSP 技术封装了许多功能, 这些功能是在易用的、与 JSP 相关的 XML 标记中生成动态内容所需要的。标准的 JSP 标记能够存取和实例化 JavaBeans 组件, 设定或者检索群组属性, 下载 Applet, 以及执行用其他方法更难于编码和耗时的功能

一种技术功能越是强大, 其复杂性就越高, JSP 技术也不例外。在使用 JSP 技术成功编写高效, 安全的 Web 网站的同时, JSP 也面临入门比较困难的问题。相对与其他网页开发技术, 如 ASP、PHP 来说, 三者各有其优点。其详细信息如表 1-3 所示。

表 1-3 JSP、PHP 和 ASP 技术比较

参数	JSP	ASP	PHP
运行速度	快	较快	较快
运行耗损	较小	较大	较大
难易程度	容易掌握	简单	简单
运行平台	绝大部分平台均可	Windows 平台	Windows/Unix 平台
扩展性	好	较好	较差
安全性	好	较差	好
函数支持	多	较少	多
数据库支持	多	多	多
厂商支持	多	较少	较多
对 XML 的支持	支持	不支持	支持
对组件的支持	支持	支持	不支持
对分布式处理的支持	支持	支持	不支持
应用程序	较广	较广	较广

此三者中, JSP 应该是未来发展的趋势。一些大的电子商务解决方案提供商都采用 JSP/Servlet。比较著名的如 IBM 的 E-business, 它的核心技术采用 JSP/Servlet 的 Web Sphere。Intershop 公司开

发的 Enfinity 软件采用 JSP/Servlet 的电子商务 Application Server, 而且他们声言不再开发传统软件。

1.1.2 JSP 的开发模式

JSP 自产生到现在, 应用越来越广泛, 其相关技术也越来越多, 如 JavaBean、EJB 等。相关技术的产生, 使 JSP 技术更容易实现 Web 网站的开发和控制。JSP 网站开发技术, 经常使用下面几种组合开发网站, 包括纯粹 JSP 技术实现、JSP+JavaBean 实现、JSP+JavaBean+Servlet 实现、J2EE 实现等。不同的开发组合, 可以称为不同的设计模式, 最常用的技术是 JSP+JavaBean+Servlet。

1. 纯粹 JSP 实现

使用纯粹 JSP 技术实现动态网站开发, 是 JSP 初学者经常使用的技术。JSP 页面中所有的代码都是在同一个页面, 如<html>标记、<css>标记、<javascript>标记、逻辑处理、数据库处理代码等。这么多代码, 混合在一个页面中, 容易出现错误, 出现错误后, 不容易查找和调试。这时设计出的网站, 采用 JSP 技术还是采用 ASP 技术就没有什么大的差别了。

2. JSP+JavaBean 实现

JSP+JavaBean 技术的使用, 很好地达到了页面静态部分和动态部分相互分离。在这种技术中, 使用 JSP 技术中的 HTML、CSS 等可以非常容易地构建数据显示页面, 而对于数据处理, 可以交给 JavaBean 技术处理, 如连接数据库代码、显示数据库代码。当执行功能代码封装到 JavaBean 中时, 同时也达到了代码重用的目的。如显示当前时间的 JavaBean, 不仅可以用在当前页面, 还可以用在其他页面。这种技术的使用, 已经显示出 JSP 技术的优势, 但并不充分, JSP+JavaBean+Servlet 技术的组合更加充分地显示了 JSP 的优势。

3. JSP+JavaBean+Servlet 实现

JSP+JavaBean+Servlet 技术的组合, 很好地实现了 MVC 模式, MVC 模式是应该提倡学习和使用的一种模式。MVC 模式是 Model-View-Controller 的缩写, 中文翻译为“模式-视图-控制器”, MVC 应用程序总是由这三个部分组成。Event (事件) 导致 Controller 改变了 Model 或 View, 或者同时改变两者。只要 Controller 改变了 Models 的数据或者属性, 所有依赖的 View 都会自动更新。类似的, 只要 Controller 改变了 View, View 会从潜在的 Model 中获取数据来刷新自己。MVC 模式最早是 Smalltalk 语言研究团提出的, 应用于用户交互应用程序。Smalltalk 语言和 Java 语言有很多相似性, 都是面向对象语言。

MVC 模式是一个复杂的架构模式, 其实现也显得非常复杂。但是, 人们已经总结出了很多可靠的设计模式, 多种设计模式结合在一起, 使 MVC 模式的实现变得相对简单易行。Views 可以看作一棵树, 显然可以用 Composite Pattern 来实现。Views 和 Models 之间的关系可以用 Observer Pattern 来实现。Controller 控制 Views 的显示, 可以用 Strategy Pattern 实现。Model 通常是一个调停者, 可采用 Mediator Pattern 来实现。

现在来了解一下 MVC 三个部分在架构中处于什么位置, 这样有助于理解 MVC 模式的实现。MVC 与架构的对应关系是: View 处于 Web Tier 或者是 Client Tier, 通常是 JSP/Servlet, 即页面

的显示部分。Controller 也处于 Web Tier，通常用 Servlet 来实现，即页面显示的逻辑部分的实现。Model 处于 Middle Tier，通常用服务器端的 JavaBean 或者 EJB 实现，即业务逻辑部分的实现。其形式如图 1-1 所示。

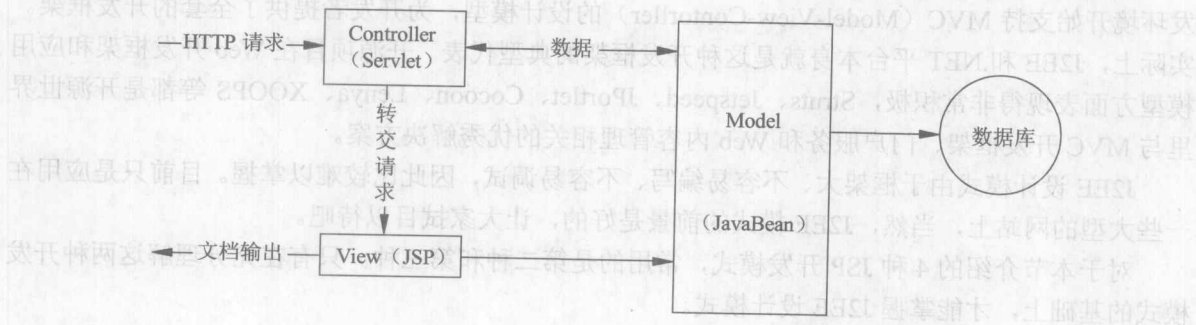


图 1-1 MVC 三个部分的形式

视图 (View) 代表用户交互界面，对于 Web 应用来说，可以概括为 HTML 界面，但有可能为 XHTML、XML 和 Applet。随着应用的复杂性和规模性，界面的处理也变得具有挑战性。一个应用可能有很多不同的视图，MVC 设计模式对于视图的处理仅限于视图上数据的采集和处理，以及用户的请求，而不包括在视图上的业务流程的处理。业务流程的处理交予模型 (Model) 来处理。比如一个订单的视图只接受来自模型的数据，并显示给用户，以及将用户界面的输入数据和请求传递给控制和模型。

模型 (Model) 就是业务流程/状态的处理，以及业务规则的制定。业务流程的处理过程对其他层来说是黑箱操作，模型接受视图请求的数据，并返回最终的处理结果。业务模型的设计可以说是 MVC 最主要的核心。通过 MVC 设计模式可知，把应用的模型按一定的规则抽取出来，抽取的层次很重要，这也是判断开发人员是否优秀的依据。抽象与具体不能隔得太远，也不能太近。

控制器 (Controller) 可以理解为从用户接受请求，将模型与视图匹配在一起，共同完成用户的请求。划分控制层的作用也很明显，它清楚地告诉人们，它就是一个分发器，选择什么样的模型，选择什么样的视图，可以完成什么样的用户请求，控制层并不做任何的数据处理。例如，用户点击一个连接，控制层接受请求后，并不处理业务信息，它只把用户的信息传递给模型，告诉模型做什么，选择符合要求的视图返回给用户。因此，一个模型可能对应多个视图，一个视图可能对应多个模型。

模型、视图与控制器的分离，使得一个模型可以具有多个显示视图。如果用户通过某个视图的控制器改变了模型的数据，所有其他依赖于这些数据的视图都应反映到这些变化。因此，无论何时发生了何种数据变化，控制器都会将变化通知所有的视图，导致显示的更新。这实际上是一种模型的变化-传播机制。

4. J2EE 模式实现

J2EE 是纯粹基于 Java 的解决方案。1998 年，Sun 发布了 EJB 1.0 标准。EJB 为企业级应用中必不可少的数据封装、事务处理、交易控制等功能提供了良好的技术基础。至此，J2EE 平台的三大核心技术 Servlet、JSP 和 EJB 都已先后问世。1999 年，Sun 正式发布了 J2EE 的第一个版本。到 2003 年时，Sun 的 J2EE 版本已经升级到了 J2EE 1.4 版，其中三个关键组件的版本也演进到了

Servlet 2.4、JSP 2.0 和 EJB 2.1。至此，J2EE 体系及相关的软件产品已经成为了 Web 服务器端开发的一个强有力的支撑环境。在这种模式里，EJB 替代了前面提到的 JavaBean 技术。

在 Web 服务器端，2000 年以后出现了几种主要的技术融合方式。首先，越来越多的 Web 开发环境开始支持 MVC（Model-View-Controller）的设计模型，为开发者提供了全套的开发框架。实际上，J2EE 和 .NET 平台本身就是这种开发框架的典型代表。开源项目在 Web 开发框架和应用模型方面表现得非常积极，Struts、Jetspeed、JPortlet、Cocoon、Lenya、XOOPS 等都是开源世界里与 MVC 开发框架、门户服务和 Web 内容管理相关的优秀解决方案。

J2EE 设计模式由于框架大、不容易编写、不容易调试，因此比较难以掌握。目前只是应用在一些大型的网站上，当然，J2EE 模式的前景是好的，让大家拭目以待吧。

对于本节介绍的 4 种 JSP 开发模式，常用的是第二种和第三种。只有在充分理解这两种开发模式的基础上，才能掌握 J2EE 设计模式。

1.2 JSP 的开发环境

在 1.1 节中，介绍了 JSP 的产生、优势和现在流行的开发模式。在了解了 JSP 的相关技术后，本节将主要介绍配置 JSP 运行环境，如 JDK 和 Tomcat 的安装。

1.2.1 安装和配置 JDK

JSP 技术目前唯一支持的脚本语言为 Java 语言，在 JSP 页面运行时，需要有相应的编译器编译和解释器来执行这些 Java 代码。执行 Java 代码，需要下载和安装 Java 的 JDK 开发工具包。JDK（Java Development Kit，Java 开发包，Java 开发工具）是一个编写 Java 的 Applet 和应用程序的程序开发环境。它由一个处于操作系统层之上的运行环境以及开发者编译、调试和运行用 Java 语言编写的 Applet 和应用程序所需的工具组成。在计算机上，安装了 JDK 开发工具包就可以编译和运行 Java 代码了。JDK 是 Sun Microsystems 针对 Java 开发人员而开发的产品。自从 Java 推出以来，JDK 已经成为使用最广泛的 Java SDK（Software Development Kit）。

1. JDK 简介

JDK 中包括 Java 编译器（javac）、打包工具（jar）、文档生成器（javadoc）、查错工具（jdb），以及完整的 JRE（Java Runtime Environment，Java 运行环境），也被称为 Private Runtime。并包括了用于产品环境的各种类库，以及给开发人员使用的补充库，如国际化的库、IDL 库。JDK 中还包括各种例子程序，用以展示 Java API 中的各部分。

从初学者的角度来看，采用 JDK 开发 Java 程序能够很快地理解程序中各部分代码之间的关系，有利于理解 Java 面向对象的设计思想。JDK 的另一个显著特点是随着 Java（J2EE、J2SE 以及 J2ME）版本的升级而升级。但它的缺点也是非常明显的，就是从事大规模企业级 Java 应用开发非常困难，不能进行复杂的 Java 软件开发，也不利于团体协同开发。

JDK 一般有三种版本，其详细信息如表 1-4 所示。

表 1-4 JDK 版本

名称	说明
SE (J2SE)	Standard Edition, 标准版, 主要用于开发 Java 桌面应用程序
EE (J2EE)	Enterprise Edition, 企业版, 使用这种 JDK 开发 J2EE 应用程序, 用于 Web 方面
ME (J2ME)	Micro Edition, 微型版, 主要用于移动设备、嵌入式设备上的 Java 应用程序

Java 开发工具 (JDK) 是许多 Java 专家最初使用的开发环境。尽管许多编程人员已经使用第三方的开发工具, 但 JDK 仍被当作 Java 开发的重要工具。JDK 由一个标准类库和一组建立、测试及建立文档的 Java 实用程序组成。API 是一些预定义类库, 开发人员需要用这些类库来访问 Java 语言的功能。Java API 包括一些重要的语言结构以及基本图形、网络和文件 I/O。一般来说, Java API 的非 I/O 部分对于运行 Java 的所有平台是相同的, 而 I/O 部分则仅在通用 Java 环境中实现。

2. 下载 JDK

获取 JDK 开发工具包非常简单, 可以直接在 JDK 官方网站下载, 也可以在网络上进行搜索。这里演示在官方网站上来获取 JDK 开发工具包, JDK 开发工具包的官方网站的网址为 <http://java.sun.com/>, 在地址栏中输入该网址, 单击【转到】按钮, 会显示如图 1-2 所示的窗口。

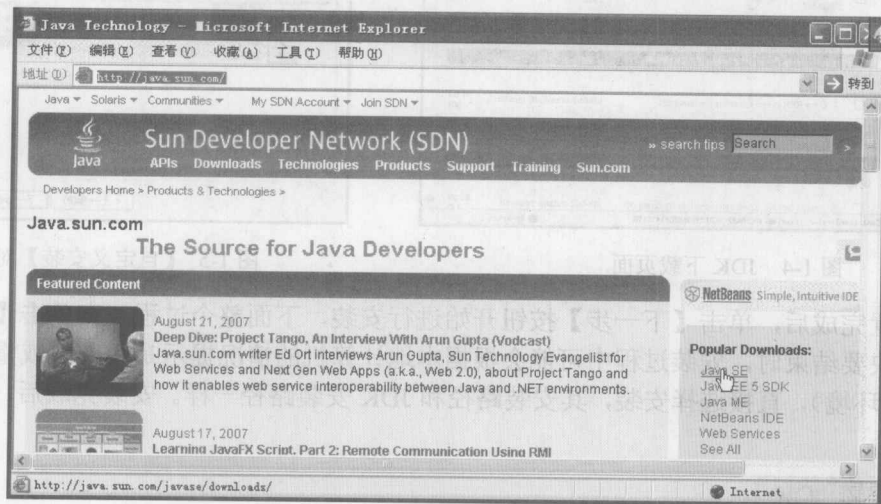


图 1-2 JDK 官方下载网站

在窗口中, 找到显示页面中的右侧, 单击 Java SE 超级链接, 会显示如图 1-3 所示的窗口。

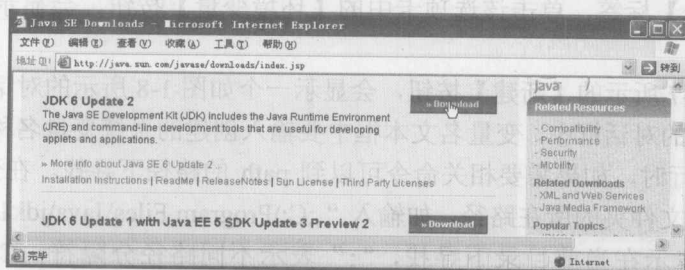


图 1-3 JDK 版本选择