

<<JSP编程及案例分析>>

图书基本信息

书名：<<JSP编程及案例分析>>

13位ISBN编号：9787512101111

10位ISBN编号：7512101112

出版时间：2010-5

出版时间：清华大学出版社

作者：郑宁 编

页数：320

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

前言

随着信息技术的发展和Internet的普及，人们越来越依赖于互联网给生活带来的便利，纷纷利用网络技术来构建自己的站点。

可以说，www技术是促进Internet高速发展的主要因素之一。

早期的网页基本上是静态网页，主要特点是“只能层现，不能交互”，当然这在互联网起步阶段起到了一定作用，但随着网络内容的膨胀，人们越来越不能满足现状，人与人之间的网络交流等都需要实时交互，而且这种需求越来越强烈。

鉴于此，各种各样的Web技术应运而生，且都是建立在一系列“活跃”的交互操作上的。

通常人们用客户—服务器这个词来描述Web。

这是一个交互的概念，一般把提出请求的一方称为客户端，而把响应请求的一方称为服务器端。

这种简单的模型是静态的，它只能对对方的激励作出响应。

而在活跃的Web中，双方都应该是活跃的，只有这样才能把客户机和服务器结合起来，产生最强的交互，这样就引出了动态网页的概念。

在Web领域里，有几十万乃至上百万的站点相互之间正在进行着激烈的竞争，它们想尽一切办法来吸引用户的注意力。

简单的、静态的页面对用户不会有太大的吸引力，只有动态的、有条理的数据加上友好的、交互性强的界面，再加上丰富的内容，才能吸引用户。

当然，数据的自动更新也非常重要。

在短短的几年时间里，Web的面貌已经发生了非常大的变化。

今天，可以在Web页面中创建应用程序、访问数据库，无论在感觉上、操作中，还是实际的用途方面都与Windows中的应用程序非常类似。

现在，商家可以与潜在客户、现有的客户、员工及其他人中的任何一个进行网络在线实时沟通，并实施一些在线的服务类的商业活动。

为了抢占Internet这个诱人的市场，在Sun公司的倡导下，许多公司共同参与建立了一种新的动态网页技术标准——Java Server Pages (JSP)。

Sun应用Java社团开发JSP。

在开发JSP规范的过程中，Sun公司与许许多多主要的Web服务器、web应用服务器和开发工具供应商，以及各种各样富有经验的开发团体进行合作，结果找到了一种适合于应用和页面开发人员的开发方法，它具有极佳的可移植性和易用性。

Sun将针对JSP的产品授权给了工具提供商（如Macromedia）、结盟公司（如Apache，Netscape）、最终用户、协作商及其他人。

最近，Sun将最新版本的ISP和JavaServlet（JSP1.1，JavaServlet2.2）的源代码发放给了Apache，以求JSP与Apache紧密结合、共同发展。

Apache、Sun和许多其他的公司及个人公开成立了一个咨询机构，以便任何公司和个人都能免费取得信息。

这样，Sun、公司就在这个领域中稳稳地站住了脚跟。

<<JSP编程及案例分析>>

内容概要

本书较为全面地介绍了JSP技术，根据作者的开发经验，按照基础篇、提高篇、数据库应用篇的次序，由浅入深、循序渐进地介绍了JSP的运行系统、基本概念、语法规范及其相关内容，并提供了大量的应用实例。

全书共分为6章，内容包括JSP概述和基础、JSP内置对象、JavaBean组件、Servlet技术、数据库操作，并结合应用实例加以巩固。

本书适合作为高职高专计算机专业的教材，也可供对Web技术感兴趣的读者自学参考。

书籍目录

第1篇 基础篇第1章 JSP概述 1.1 JSP简介 1.1.1 什么是JSP 1.1.2 与Java Servlet的关系 1.1.3 JSP技术特点 1.1.4 与其他技术比较 1.2 图解开发环境 1.2.1 JDK安装及配置 1.2.2 Tomcat安装及配置 1.2.3 建立虚拟工作目录 1.2.4 MyEclipse安装及配置 1.3 语法介绍 1.3.1 JSP页面中的元素 1.3.2 JSP语法概要 1.3.3 模板文本(静态HTML) 1.4 运行第一个JSP程序 1.5 入门实例 1.5.1 实例1: HTML练习 1.5.2 实例2: 显示日期时间 1.6 本章小结 1.7 习题第2章 JSP基础 2.1 JSP基本语法 2.1.1 JSP声明 2.1.2 JSP表达式 2.1.3 脚本段 2.2 JSP的指令 2.2.1 page指令 2.2.2 include指令 2.2.3 taglib指令 2.3 JSP的动作 2.3.1 jsp:include动作 2.3.2 jsp:useBean动作 2.3.3 jsp:setProperty动作 2.3.4 jsp:getProperty动作 2.3.5 jsp:forward动作 2.3.6 jsp:plugin动作 2.3.7 jsp:param动作 2.4 JSP的注释 2.5 JSP基础实例 2.5.1 实例1: JSP中方法定义 2.5.2 实例2: 计算任意阶乘 2.5.3 实例3: JSP的出错处理 2.5.4 实例4: 使用forward动作 2.5.5 实例5: 使用param动作 2.5.6 实例6: 使用include动作 2.6 本章小结 2.7 习题第3章 JSP内置对象 3.1 request对象 3.1.1 request对象常用方法 3.1.2 表单信息读取 3.1.3 request乱码问题 3.2 response对象 3.2.1 response对象常用方法 3.2.2 动态响应contentType属性 3.2.3 response的HTTP文件头 3.2.4 response重定向 3.2.5 response的状态行 3.3 session对象 3.3.1 session对象常用方法 3.3.2 session对象的ID 3.3.3 session对象与URL重写 3.3.4 session对象的使用 3.4 application对象 3.5 out对象 3.5.1 out对象常用方法 3.5.2 out对象之print()和println()对比 3.6 exception对象 3.6.1 exception对象常用方法 3.6.2 exception对象的使用 3.6.3 运行中可能出现的问题 3.7 其他内置对象 3.8 JSP内置对象实例 3.8.1 实例1: 防刷新计数器 3.8.2 实例2: 用户注册信息 3.8.3 实例3: 获取请求相关信息 3.8.4 实例4: 表单信息读取 3.8.5 实例5: 提交表单计算平方根 3.8.6 实例6: 中文乱码问题 3.8.7 实例7: 动态响应contentType属性 3.8.8 实例8: 定时刷新页面 3.8.9 实例9: response重定向 3.8.10 实例10: 修改response状态码 3.8.11 实例11: session与URL重写 3.8.12 实例12: session的ID 3.8.13 实例13: session对象的使用 3.8.14 实例14: application对象的使用 3.8.15 实例15: out对象的print()与println()对比 3.8.16 实例16: out对象的使用 3.8.17 实例17: exception对象的使用 3.9 本章小结 3.10 习题第2篇 提高篇第4章 JavaBean组件 4.1 什么是JavaBean 4.2 编写和使用JavaBean 4.2.1 编写JavaBean 4.2.2 使用JavaBean 4.3 获取和设置JavaBean属性 4.3.1 获取JavaBean属性 4.3.2 设置JavaBean属性 4.4 JavaBean应用实例 4.4.1 实例1: 编写JavaBean 4.4.2 实例2: 使用JavaBean 4.4.3 实例3: 获取JavaBean属性 4.4.4 实例4: 设置JavaBean属性 4.4.5 实例5: 简单计算器 4.4.6 实例6: 彩色验证码 4.4.7 实例7: 文本的读写 4.4.8 实例8: 猜数字游戏 4.5 本章小结 4.6 习题第5章 Servlet技术 5.1 Servlet简介 5.1.1 什么是Servlet 5.1.2 Servlet基本结构 5.1.3 Servlet的映射 5.1.4 Servlet生命周期 5.2 初始化参数 5.2.1 Servlet初始化参数 5.2.2 实例: 测试Servlet初始化参数 5.2.3 ServletContext初始化参数 5.2.4 实例: 测试ServletContext初始化参数 5.3 Servlet请求分派 5.3.1 请求分派 5.3.2 重定向与请求分派 5.3.3 实例: 重定向/请求分派 5.4 理解会话 5.4.1 会话管理机制 5.4.2 会话对象中读取/保存数据 5.5 Servlet过滤器 5.5.1 过滤器工作原理 5.5.2 Servlet过滤器API 5.5.3 过滤器相关接口工作流程 5.5.4 过滤器配置 5.5.5 过滤器链配置 5.5.6 实例1: 简单过滤器 5.5.7 实例2: 过滤器链 5.6 Servlet监听器 5.6.1 接口及相关事件类 5.6.2 监听器配置 5.6.3 实例1: 上下文相关监听器应用 5.6.4 实例2: HttpSessionListener应用 5.6.5 实例3: HttpSessionAttribute-Listener应用 5.6.6 实例4: HttpSessionBinding Listener应用 5.6.7 实例5: 请求相关监听器应用 5.7 Servlet上下文 5.8 MVC迷你教程 5.8.1 MVC模式介绍 5.8.2 实战MVC 5.9 Servlet实例 5.9.1 实例1: 手工完成Servlet程序 5.9.2 实例2: 用户登录验证 5.9.3 实例3: Web定时器 5.9.4 实例4: 调查反馈 5.9.5 实例5: 会话管理 5.9.6 实例6: 重定向与请求分派 5.10 本章小结 5.11 习题第3篇 数据库应用第6章 JSP数据库操作 6.1 数据库基础 6.1.1 关系数据库简介 6.1.2 Access数据库 6.1.3 SQL Server数据库 6.1.4 Oracle数据库 6.1.5 MySQL数据库 6.2 JDBC简介 6.2.1 什么是JDBC 6.2.2 JDBC的用途 6.2.3 JDBC的类型 6.2.4 JDBC的入门 6.3 JSP中使用数据库 6.3.1 数据库的连接过程 6.3.2 几种数据库的连接 6.4 数据操作 6.4.1 添加记录 6.4.2 查询记录 6.4.3 更新记录 6.4.4 删除记录 6.5 JSP数据库应用实例 6.5.1 实例1: 留言板 6.5.2 实例2: 通讯录 6.6 本章小结 6.7 习题附录A 留言板系统 A1 数据存储 A2 Java程序代码

章节摘录

为了克服Servlet的缺点，Sun公司以JavaServlet为基础，推出了JSP。

JSP具有Servlet的几乎所有的优点。

当一个客户请求一个JSP页面时，JSP引擎根据JSP页面生成一个Java文件，即一个Servlet。

用JSP支持Java Bean这一特点，可以有效地管理页面的逻辑部分和页面的输出部分。

另外，JSP也可以和Servlet有效地结合，分离页面的逻辑部分和页面的输出部分。

1.1.3 JSP技术特点 关于JSP技术特点，主要有以下几点。

1.将内容的生成和显示进行分离 使用JSP技术，Web页面开发人员可以使用HTML或者XML标识来设计和格式化最终页面，使用JSP标识或者小脚本来生成页面上的动态内容。

生成内容的逻辑被封装在标识和Java Bean组件中，并且捆绑在小脚本中，所有的脚本在服务器端运行。

如果核心逻辑被封装在标识和Java Bean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，而不影响内容的生成。

在服务器端，JSP引擎解释JSP标识和小脚本，生成所请求的内容（如通过访问Java Bean组件，使用JDBC技术访问数据库或者包含文件），并且将结果以HTML（或者XML）页面的形式发送回浏览器。

这既有助于保护自己的代码，又可保证任何基于HTML的Web浏览器的完全可用性。

2.一次编译。

到处运行 由于JSP页面的内置脚本语言基于Java语言，而所有的JSP页面都要被编译成为Servlet，这样JSP页面就具有Java技术的所有优点，包括健壮的存储管理和安全性等。

当然其中最重要的一点就是“一次编译，到处运行”。

JSP技术是与设计平台完全无关的，包括它的动态Web页面、Web服务器和底层的服务器组件。

你可以在任何平台上编写JSP页面，在任何Web服务器或者Web应用服务器上运行，或者通过任何Web浏览器访问。

有了这个优点，随着越来越多的供应商将JSP支持添加到他们的产品中，你就可以使用自己所选择的服务器和工具了。

更改工具或服务器并不会影响到当前的应用。

3.强调可重用的组件 绝大多数JSP页面依赖于可重用的、跨平台的组件（JavaBean或者企业版的JavaBean组件）来执行应用程序中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客户团体所使用。

这些组件有助于将网页的设计与逻辑程序的编写分开，节约了开发时间，同时充分发挥了Java和其他脚本语言的跨平台的能力和灵活性。

基于组件的方法加速了总体开发过程，并且使得各种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。

4.采用标记简化页面的开发 Web页面开发人员不一定是熟悉脚本语言的编程人员。

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标记中进行动态内容生成时所必需的。

<<JSP编程及案例分析>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>