



- ✓ 深入剖析JSP+Ajax技术要点和难点
- ✓ 围绕丰富实例讲解JSP+Ajax的Web 2.0开发实践知识
- ✓ 精心编写大量“实验指导”，引导学生深入学习编程实践
- ✓ 课后提供丰富习题，巩固学习成果
- ✓ 网站提供代码下载和课件支持



■ 张银鹤 刘治国 张豪 等编著

JSP动态网站开发

实践教程(第2版)

清华大学出版社





- ✓ 深入剖析JSP+Ajax技术要点和难点
- ✓ 围绕丰富实例讲解JSP+Ajax的Web 2.0开发实践知识
- ✓ 精心编写大量“实验指导”，引导学生深入学习编程实践
- ✓ 课后提供丰富习题，巩固学习成果
- ✓ 网站提供代码下载和课件支持



■ 张银鹤 刘治国 张豪 等编著

JSP动态网站开发

实践教程(第2版)

清华大学出版社
北 京

内 容 简 介

本书介绍 Web 2.0 时代 Ajax 和 JSP 动态网站开发技术。全书分为 11 章, 介绍 Web 2.0 的概念、JSP 和 Ajax 基础知识, JSP 页面元素, JSP 内置对象, JavaBean 技术, Servlet 技术, JSP 访问数据库, JSP 处理 XML 文件, Ajax 的各种相关技术 (如 CSS、JavaScript、DOM 和 XMLHttpRequest 等), Ajax 常用技巧, Ajax 框架等内容。本书最后一章通过综合实例, 演示了 Web 2.0 下的 Ajax+JSP 技术的整合。本书提供了实例源代码和教学视频文件由清华大学出版社网站提供。

本书可作为普通高等院校计算机专业和非计算机专业的 JSP 动态网站开发教材, 也可作为 Web 2.0 项目开发人员的自学参考用书。

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

JSP 动态网站开发实践教程 (第 2 版) / 张银鹤等编著. —北京: 清华大学出版社, 2009. 5
ISBN 978-7-302-19423-1

I. J… II. 张… III. JAVA 语言—主页制作—程序设计—教材 IV. TP393.092

中国版本图书馆 CIP 数据核字 (2009) 第 013319 号

责任编辑: 夏兆彦

责任校对: 徐俊伟

责任印制: 王秀菊

出版发行: 清华大学出版社

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

社 总 机: 010-62770175

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

地 址: 北京清华大学学研大厦 A 座

邮 编: 100084

邮 购: 010-62786544

印 刷 者: 北京嘉实印刷有限公司

装 订 者: 三河市兴旺装订有限公司

经 销: 全国新华书店

开 本: 185×260 印 张: 27.25 字 数: 654 千字

版 次: 2009 年 5 月第 1 版 印 次: 2009 年 5 月第 1 次印刷

印 数: 1~5000

定 价: 39.80 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: 031600-01

JSP (Java Server Pages) 是由 Sun Microsystems 公司倡导、许多公司参与一起建立的一种动态网页技术标准。JSP 本身所具有的优势是其他 Web 技术所不能具有的。Web 2.0 以 XML、RSS 和 Ajax 等技术为基础, 融合了 Web 1.0 的应用模式及新出现的 Blog、SNS 和 Wiki 等多种多对多互动应用服务模式。Ajax 是 Web 2.0 的一项关键技术, 它允许把用户和 Web 页面间的交互与 Web 浏览器和服务器间的通信分离开来, 进行异步传送数据。Ajax 的出现使瘦服务器端成为了现实。

在 Web 2.0 时代, 使用 JSP 技术开发网站, 不仅仅需要关注服务器端程序的编写, 还要关注客户端技术的使用, 以加强用户的体验。Web 2.0 时代的编程, 不再是一种 JSP 服务器技术, 而是客户端技术和服务器技术的整合。本书所介绍的 JSP 技术, 已不是纯粹的 JSP 技术, 而是一个范围更大的服务器端 JSP 技术。

1. 本书主要内容

本书首先对 Web 2.0 进行详细的概述, 并简单介绍了本书所涉及的两种技术——Ajax 和 JSP。接下来通过一个 Web 2.0 应用演示了服务器和浏览器之间的交互过程, 让读者对 JSP 和 Ajax 技术在 Web 2.0 应用中扮演的角色有更深入的认识。本书主要内容如下。

第 1 章 Web 2.0 下的 JSP 和 Ajax, 首先介绍 Web 2.0 的概念、JSP 的概念和 Ajax 的概念, 接下来介绍 JSP+Ajax 结合的意义, 然后介绍 JSP 的运行环境和开发环境, 最后以案例的方式介绍 Ajax+JSP 的使用。

第 2 章 JSP 页面元素, 介绍 JSP 页面的各个组成元素, 如脚本元素、指令元素和动作元素。

第 3 章 JSP 内置对象, 介绍 JSP 常用的内置对象, 如请求对象 request、响应对象 response 和会话对象 session 等。

第 4 章 JavaBean 技术, 详细介绍 JavaBean 的定义、调用, 并在最后以实例的方式演示了 JavaBean 的使用。

第 5 章 Servlet 技术, 首先介绍 Servlet 的概念、创建和使用, 然后介绍 JSP 访问 Servlet 的方式, 接下来介绍 doGet 和 doPost 方法的应用、重定向和转发, 最后介绍会话管理等内容。

第 6 章 JSP 访问数据库, 介绍 JDBC 的基本概念和相关接口, 以及如何使用 JDBC 接口操作数据库, 使用预编译语句等。

第 7 章 JSP 处理 XML, 介绍 XML 的概念、语法、约束文档 DTD 等基本概念, 以及利用 DOM 对 XML 文档进行访问和更新 XML 文档。

第 8 章 Ajax 概述, 首先介绍 Ajax 的基本原则和原理, 并对传统 Web 应用解决方案进行分析, 最后以大量篇幅介绍了 Ajax 的各种相关技术, 如 CSS、JavaScript、DOM 和

XMLHttpRequest 等。

第 9 章 Ajax 常用技巧, 展示了一些常用和经典的实例, 并介绍了 Ajax 技术常用实现技巧。如滚动新闻、局部更新等。

第 10 章 Ajax 框架, 详细介绍了 Ajax 的框架, 首先介绍现存的 Ajax 框架, 并重点介绍 3 个比较流行的 Ajax 框架, 即 Prototype、DWR 和 Dojo 框架, 对于每种框架, 都有相应的实例。

第 11 章投票管理系统, 介绍投票系统实现的过程, 如系统的需求分析、系统设计、数据库设计和模块实现等。演示了 Web 2.0 下的 Ajax+JSP 技术的整合。

2. 本书主要特色

本书通过实例, 结合 Ajax 介绍 JSP 动态网站开发知识, 具有实用性教程的特色。

- 本书汇总了作者多年的程序员职业教学培训经验, 内容组织合理, 实例丰富全面。
- 本书使用 JSP 语言开发了大量实例, 可以通过这些丰富的实例学习 JSP 编程实践知识。
- 本书编写了大量“实验项目”, 引导读者应用该章知识独立练习编程项目。
- 每章课后练习题帮助读者检查对 JSP 开发理论知识的掌握程度。
- 清华大学出版社网站提供本书实例完整源代码和教学视频文件。

3. 本书读者对象

书中采用大量的实例进行讲解, 力求通过实例使读者更形象地理解面向对象思想, 快速掌握 JSP 开发技术。本书难度适中, 内容由浅入深, 实用性强。每章附有精心编写的实验和习题, 便于实践和巩固所学知识。本书可作为普通高等院校 JSP 程序设计课程的教材, 也可作为 Web 2.0 项目开发人员的参考用书。

除了封面署名人员外, 参与本书编写的还有李乃文、肖新峰、宋强、马海军、许勇光、王泽波、孙江玮、田成军、刘俊杰、李海庆、王树兴、朱俊成、王敏、张瑞萍、王黎、安征、亢凤林、康显丽、李海峰、崔群法、孙岩、祁凯、倪宝童、王立新、吴越胜、何方等。

由于时间仓促, 作者水平有限, 疏漏之处在所难免, 欢迎读者朋友登录清华大学出版社的网站 www.tup.com.cn 与我们联系, 帮助我们改进提高。

第 1 章 Web 2.0 下的 JSP 和 Ajax1

- 1.1 Web 2.0 与 JSP、Ajax 简介1
 - 1.1.1 Web 2.0 简介1
 - 1.1.2 JSP 概述3
 - 1.1.3 Ajax 概述4
- 1.2 JSP+Ajax 结构及其意义5
 - 1.2.1 Web 应用解决方案5
 - 1.2.2 Ajax 解决方案的优势7
- 1.3 构建 JSP 运行环境7
 - 1.3.1 安装 JDK 开发工具包8
 - 1.3.2 安装 Tomcat 服务器11
 - 1.3.3 记事本开发 JSP 实例15
- 1.4 构建 JSP 开发环境17
 - 1.4.1 安装 Eclipse17
 - 1.4.2 安装 MyEclipse20
 - 1.4.3 Eclipse 开发 JSP 实例25
- 1.5 JSP 与 Servlet 的关系29
- 1.6 Ajax+JSP 实例30
- 1.7 实验指导31
 - 1.7.1 使用 HTML31
 - 1.7.2 使用 CSS33
 - 1.7.3 使用 JavaScript35
 - 1.7.4 构建 JSP 环境36
- 1.8 习题集38

第 2 章 JSP 页面元素40

- 2.1 JSP 页面基本结构40
- 2.2 JSP 注释42
- 2.3 脚本元素44
 - 2.3.1 JSP 表达式44
 - 2.3.2 JSP 脚本45
 - 2.3.3 JSP 声明46
- 2.4 指令元素49

- 2.4.1 page 指令49
- 2.4.2 include 指令51
- 2.5 动作元素53
 - 2.5.1 <jsp:include>包含动作53
 - 2.5.2 <jsp:forward>转向动作55
 - 2.5.3 <jsp:param>传递参数指令58
 - 2.5.4 执行 Applet 或 Bean
 标签<jsp:plugin>59

- 2.6 实验指导61
 - 2.6.1 输出时间和问候61
 - 2.6.2 求阶乘63
 - 2.6.3 include 的应用65
- 2.7 习题集66

第 3 章 JSP 内置对象71

- 3.1 request 请求对象71
 - 3.1.1 request 对象概述及方法71
 - 3.1.2 获取客户提交信息73
 - 3.1.3 处理汉字信息75
 - 3.1.4 获取客户端基本信息78
- 3.2 response 响应对象80
 - 3.2.1 response 对象概述及方法80
 - 3.2.2 response 重定向81
 - 3.2.3 response 的 HTTP 文件头84
 - 3.2.4 动态响应 contentType
 属性87
- 3.3 session 会话对象88
 - 3.3.1 session 对象概述及方法89
 - 3.3.2 session 对象存储数据90
- 3.4 application 全局对象92
 - 3.4.1 application 对象
 概述及方法92

3.4.2 用 application 制作 网页计数器	93	提交数据	144
3.5 out 输出对象及其他对象	94	5.3.2 通过超链接访问 Servlet	146
3.6 实验指导	97	5.4 doGet 和 doPost 方法	149
3.6.1 用户登录	97	5.5 重定向与转发	153
3.6.2 使用 session 购物车	98	5.5.1 重定向 sendRedirect 方法	153
3.6.3 实现页面计数器	101	5.5.2 RequestDispatcher 转发对象	155
3.7 习题集	102	5.6 会话管理	161
第 4 章 JavaBean 技术	104	5.6.1 会话介绍	161
4.1 JavaBean 介绍	104	5.6.2 获取用户会话	162
4.1.1 定义 JavaBean	104	5.7 config 实现简单计数器	166
4.1.2 JavaBean 规则	106	5.8 实验指导	168
4.1.3 简单的 JavaBean 例子	107	5.8.1 读取数据库数据	168
4.2 使用 JavaBean	107	5.8.2 Servlet 实现控制	170
4.3 JavaBean 作用域范围	110	5.8.3 实现网上购物	172
4.3.1 page 作用域	110	5.9 习题集	174
4.3.2 request 作用域	112	第 6 章 JSP 访问数据库	178
4.3.3 session 作用域	114	6.1 JDBC 概述	178
4.3.4 application 作用域	115	6.2 JDBC API 介绍	180
4.4 获取和设置属性	117	6.3 连接数据库的常用方式	185
4.4.1 setProperty 动作标记	117	6.3.1 使用纯 Java 数据库驱动 程序连接	185
4.4.2 getProperty 动作标记	121	6.3.2 使用 JDBC-ODBC 桥 接器连接	188
4.5 Javabean 辅助类	122	6.4 更新、添加与删除操作	191
4.6 JavaBean 实例: 四则运算 bean	123	6.5 记录显示和查询操作	194
4.7 实验指导	126	6.5.1 数据显示	194
4.7.1 实现日历	126	6.5.2 模糊查询	196
4.7.2 实现留言板	128	6.6 分页显示记录	198
4.7.3 实现购物车	130	6.7 使用预处理语句	201
4.8 习题集	132	6.7.1 预处理语句的优点	201
第 5 章 Servlet 技术	135	6.7.2 预处理语句应用	202
5.1 Servlet 对象	135	6.8 实验指导	204
5.1.1 Servlet 概述	135	6.8.1 使用 JDBC 创建数据库、 表和字段	204
5.1.2 HttpServlet 类	136	6.8.2 配置 MySQL 数据库 连接池	205
5.1.3 部署和运行 Servlet	138		
5.2 Servlet 工作原理	141		
5.3 通过 JSP 页面调用 Servlet	144		
5.3.1 通过表单向 Servlet			

6.8.3 JSP 通讯录	207	8.1 理解 Ajax	273
6.9 习题集	211	8.1.1 Ajax 相关技术简介	273
第 7 章 JSP 处理 XML	214	8.1.2 Ajax 工作原理	275
7.1 XML 技术概述	214	8.1.3 Ajax 的基本原则	276
7.1.1 XML 技术产生背景	214	8.2 CSS 样式	278
7.1.2 XML 技术概述	215	8.2.1 概述	279
7.1.3 XML 简单例子	216	8.2.2 基本语法格式	279
7.2 XML 语法	217	8.2.3 引入样式表	283
7.2.1 XML 文档结构	217	8.2.4 常用样式属性	284
7.2.2 XML 文档元素	219	8.3 JavaScript 脚本语言	290
7.2.3 XML 命名空间	224	8.3.1 JavaScript 简介	291
7.3 DTD	227	8.3.2 JavaScript 基本语法	292
7.3.1 DTD 概念	227	8.3.3 JavaScript 函数	294
7.3.2 DTD 声明	230	8.3.4 JavaScript 对象	296
7.4 XML 文档解析	234	8.3.5 JavaScript 事件	300
7.5 DOM 介绍	235	8.4 JavaScript 操作 DOM	302
7.6 DOM 解析接口和对象	237	8.4.1 JavaScript 获取 XML 文档数据	302
7.7 DOM 访问节点	240	8.4.2 JavaScript 更新 XML 文档	303
7.7.1 DOM 对象的创建 及调用	240	8.4.3 JavaScript 操作 HTML 文档	306
7.7.2 访问 Document 节点	243	8.5 XMLHttpRequest	307
7.7.3 访问 Element 节点	245	8.5.1 XMLHttpRequest 简介	307
7.7.4 访问 Text 节点	247	8.5.2 XMLHttpRequest 属性 和方法	308
7.7.5 访问 Attr 节点	250	8.6 Ajax 实例	309
7.8 DOM 对文档操作	252	8.7 实验指导	311
7.8.1 动态创建 XML 文档	252	8.7.1 “HelloWord” 实例	311
7.8.2 添加和修改文本节点	256	8.7.2 客户端解析普通字符串	313
7.9 实验指导	258	8.7.3 客户端解析 XML 格式 字符串	315
7.9.1 使用命名空间	258	8.8 习题集	317
7.9.2 使用 DTD 规范 XML 文档	259	第 9 章 Ajax 常用技巧	322
7.9.3 JSP 利用 DOM 解析 XML	261	9.1 客户端以表格形式显示 XML 数据	322
7.9.4 DOM 添加与删除 元素节点	263	9.2 客户端解析服务器自动生成 XML 文件	325
7.10 习题集	265		
第 8 章 Ajax 概述	273		

9.3 实现新闻滚动播放	328	10.4.4 Dojo 框架实现提示 对话框	381
9.4 实现自动刷新页面	330	10.4.5 Dojo 事件处理	382
9.5 实现 Web 页面的局部动态更新	331	10.4.6 Dojo 异步调用	385
9.6 读取响应头部	334	10.4.7 Dojo 框架实现进度条	386
9.7 无刷新写入文本	336	10.5 实验指导	389
9.8 实现工具提示	338	10.5.1 使用 Dojo 框架	389
9.9 切换标签	341	10.5.2 使用 Prototype 框架	392
9.10 邮编自动完成	343	10.5.3 使用 DWR 框架	393
9.11 实验指导	345	10.6 习题集	395
9.11.1 用户登录	345		
9.11.2 自动保存草稿	348		
9.11.3 动态加载下拉列表框	350		
9.12 习题集	352		
第 10 章 Ajax 框架	358	第 11 章 投票管理系统	399
10.1 Ajax 框架概述	358	11.1 系统概述	399
10.2 Prototype 框架	360	11.1.1 需求分析	399
10.2.1 Prototype 框架简介	361	11.1.2 系统用例图	400
10.2.2 Prototype 库介绍	362	11.1.3 系统设计	402
10.2.3 Prototype 开发	366	11.2 数据库设计	403
10.3 DWR 框架	368	11.3 系统功能模块设计	404
10.3.1 DWR 框架概述	368	11.3.1 投票模块	404
10.3.2 部署 DWR 框架	369	11.3.2 投票管理模块	405
10.3.3 DWR 框架函数	372	11.3.3 投票人信息管理模块	407
10.3.4 身份验证	374	11.4 模块功能实现	407
10.4 Dojo 框架	376	11.4.1 创建项目	408
10.4.1 Dojo 框架概述	376	11.4.2 通用模块实现	408
10.4.2 安装 Dojo	378	11.4.3 投票模块实现	411
10.4.3 Dojo 自定义组件的使用	379	11.4.4 投票统计实现	419
		11.4.5 投票主题管理实现	422
		11.4.6 投票选项管理实现	428

第1章 Web 2.0 下的 JSP 和 Ajax



内容摘要 | Abstract

Web 2.0 时代的个性化不是原来的网页个性化,而是服务个性化,人们面对的将是一个全新的个性化互联网。随之而来的就是 Web 2.0 时代的 Web 程序编程也要发生变化,Ajax 的出现改变了传统 Web 编程的模式。大量的工作正在悄悄地移植到客户端,Web 2.0 时代的 Web 程序正在以新的方式出现。而 JSP 和 Ajax 技术的结合,是跨平台和负载平衡的结合。

本章将首先介绍 Web 2.0、JSP 和 Ajax 的概念,接下来介绍 JSP+Ajax 结合的意义,然后介绍 JSP 的运行环境和开发环境,最后以案例的方式介绍 Ajax+JSP 的使用。



学习目标 | Objective

- 理解 Web 2.0 的产生意义和特点
- 掌握 JSP 的概念
- 掌握 Ajax 的概念
- 熟练掌握构建 JSP 运行和开发环境的方法
- 连接 JSP 和 Servlet 的区别
- 掌握开发 Ajax+JSP 案例流程

1.1 Web 2.0 与 JSP、Ajax 简介

自从 Web 2.0 的概念提出以来,经过几年的发展,Web 2.0 已经成为互联网行业最受关注的词汇。Web 2.0 是一种以 XML、RSS、Ajax 等技术为基础,融合了 Web 1.0 的应用模式及新出现的 Blog、SNS、Wiki 等多种多对多互动应用服务模式,来满足不同用户社会化、人性化需求的服务平台。

Web 2.0 的出现改变了传统的 Web 开发模式。在传统的 Web 开发中,可以使用 JSP、PHP 和 ASP 中的任一种 Web 技术来构建 Web 程序,但它们对于 Web 客户端技术都是鞭长莫及,导致大量的工作都是在服务器端完成,客户端仅仅只是显示数据的层面,从而形成了胖服务器端和瘦客户端的现象。而 Web 2.0 技术的出现,改变了这种状况。可以利用 Ajax 技术在客户端创建一个中间层,将服务器端的一些工作移植到客户端。



1.1.1 Web 2.0 简介

2001 年秋,因特网公司(dot-com)泡沫的破灭标志着因特网的一个转折点。许多人断定因特网被过分炒作,事实上网络泡沫和相继而来的股市大衰退看起来像是所有技

术革命的共同特征。股市大衰退通常标志着蒸蒸日上的技术已经开始占领中央舞台。假冒者被驱逐，而真正成功的 IT 技术将展示它们的力量，同时人们开始理解是什么将一个技术同另外一个区分开来。

Web 2.0 的概念产生于 O'Reilly 公司和 MediaLive 国际公司的会议讨论中。因特网先驱和 O'Reilly 公司副总裁戴尔·多尔蒂（Dale Dougherty）注意到，同所谓的“崩溃”迥然不同，因特网比其他任何时候都更重要，令人激动的新应用程序和网站正在以令人惊讶的规律性涌现出来。更重要的是，那些幸免于当初网络泡沫的公司，看起来有一些共同之处。那么会不会是因特网公司那场泡沫的破灭标志了因特网的一种转折，以至于呼吁 Web 2.0 的行动有了意义？人们都认同这种观点，Web 2.0 的概念由此诞生。

Web 本来也没有版本之分，直到近来业界的创业者将以前的技术和应用重新“包装”起来，并贴上了软件行业的版本术语标签，Web 2.0 才大行其道了。Web 2.0 是否应该叫 Web 2.0 这并不重要，重要的是现在的 Web 2.0 的概念。Web 2.0，是相对 Web 1.0（2003 年以前的因特网模式）的新的一类因特网应用的统称，是一次从核心内容到外部应用的革命。由 Web 1.0 单纯通过网络浏览器浏览 html 网页模式向内容更丰富、联系性更强、工具性更强的 Web 2.0 因特网模式的发展，已经成为因特网新的发展趋势。

Web 1.0 到 Web 2.0 的转变，具体地说，是从模式上单纯的“读”向“写”、“共同建设”发展；由被动地接收因特网信息向主动创造因特网信息迈进。从基本构成单元上，是由“网页”向“发表/记录的信息发展”；从工具上，是由因特网浏览器向各类浏览器、rss 阅读器等发展；运行机制上，由 Client Server 向 Web Services 转变；作者由程序员等专业人士向全部普通用户发展；应用上由初级的应用向全面大量应用发展。

关于 Web 1.0 和 Web 2.0，可以从下面几个方法论述。

□ 技术和规范

Web 1.0 版本是基于 HTML 规范的网页，而 Web 2.0 是基于 RSS/ATOM 规范（XML 与 HTML 混合规范）的网页。Web 的基础协议是 HTTP（超文本传输协议），HTML 规范是 HTTP 协议中的一种应用。在因特网的最初，正是由于 HTML 使得各种各样的多媒体信息（图片、文字、影音）集成到一个网页单元中显示，互联网得以丰富多彩地发展。然而，HTML 是一种面向格式的规范，只是单纯的描述如何显示信息，是完全的非结构化数据集合。这让人类去阅读当然没有问题，但假如让程序去阅读，那是件痛苦的事。为了解决 Web 上的计算机进行数据交换的问题，完整的结构化数据规范 XML 产生了。

XML 的产生确实是好事，但 XML 的复杂性和严密性，令众多的应用开发者望而生畏，对于某些简单发布的应用简直就是杀鸡用牛刀。对于非结构化内容（文本图片等）的发布，人们选择了较为简单的规范——RSS。这种在 HTML 向 XML 过渡的中间产物，当然也可以理解为 XML 在 Web 显示界面上的应用，它在本质上确实是 XML 和 HTML 规范的混合体，离开了 Web 浏览器，RSS 根本无法很好地表现自己（至少现在的 RSS 版本是如此）。至于 Ajax 等不用提交整个网页就可以更新网页的某一部分内容的功能，其本质就是 XML 和 JavaScript。

□ Web 应用

Web 1.0 是由网站提供服务和发布内容，让浏览者阅读。Web 2.0 是由网站提供服务，让网站用户发布及浏览。Web 2.0 中，由浏览者到用户角色的转变或许是 Web 2.0 概念中

最有争议的一点,在过去的 Web 应用中难道就没有现在 Web 2.0 的应用吗? Blog 在过去是个人主页,交友网站、社区网站、BBS 论坛等都是 Web 1.0 时代的产物,甚至连 RSS 也是 Web 1.0 时代产生的应用。Web 2.0 在本质上和 Web 1.0 没有区别,与其说 Web 2.0 是 Web 1.0 的后一版本,不如说 Web 2.0 只是 Web 1.0 众多应用中关于用户交互服务的一个分类、一个方向。

□ 中心化与去中心化

Web 是基于 HTTP 协议的,也就是一个基于 B/S 模式的应用,由这种 B/S 模式所决定的必然是中心化的服务体系,无论是 Web 1.0 还是 Web 2.0 必然都不能离开一个服务提供中心。那么很多人谈论的去中心化究竟是什么不再中心化呢?笔者可以明确说明,是信息源去中心化了,而信息服务依然由一个中心向众多的用户提供服务。随着众多数码产品的普及和宽带接入成本的下降,信息源发布的成本将会越来越低,这就使得信息发布源越来越多,这是信息科技进步的结果。在以前,要让自己的言论、观点被亿万人了解是代价高昂的,需要花费几万元在报纸上刊登自己的广告。即使在因特网初期,由于服务器硬件成本的昂贵和网络接入带宽的高价,还有技术实现的高难度都让信息发布的成本高昂。而现在因特网发展了,在规模经济下,人们不必再担心成本和技术的问题了,信息采集设备(数码相机、数字录制设备、影像记录设备)和发布工具的普及,使得信息发布成本非常低廉,自然而然地让以前 Web 1.0 的浏览者也尝试体验起信息发布者的角色。这就使得以前信息采集和信息源转变为去中心化。

其实,Web 2.0 的概念是明显针对门户媒体发布模式而提出的。因为因特网还有许许多多的用户只是为了看看新闻,获得一下资讯,在很多人的意识中因特网的 Web 应用就是浏览信息。然而,Web 的应用是广泛的,真正懂得 Web 的 IT 人会把 Web 当作软件来用,从这个角度来说,似乎比那些只会阅读的浏览者高版本了,所以就有了 Web 1.0 和 Web 2.0。然而,Web 其实还是那个 Web,根本就没有版本的区别,Web 2.0 只是想某些 Web 应用和用户在因特网 Web 应用中细分出来而已。

1.1.2 JSP 概述

通过 Web 技术,实现了因特网上资源的共享。如可以在计算机上使用网页把信息或者数据显示出来,在异地的计算机可以通过因特网访问这些资源。Web 是一种典型的分布式应用架构。Web 应用中的每一次信息交换都要涉及到客户端和服务端两个层面。因此,Web 开发技术大体上也可以被分为客户端技术和服务端技术两大类。

Web 服务端的开发技术由静态向动态逐渐发展、完善起来。最早的 Web 服务器简单地响应浏览器发来的 HTTP 请求,并将存储在服务器上的 HTML 文件返回给浏览器。这种静态生成网页的技术,远远不能满足用户的需求。1993 年,CGI 1.0 的标准草案产生,该草案由 NCSA (National Center for Supercomputing Applications) 提出。CGI 技术允许服务端的应用程序根据客户端的请求,动态生成 HTML 页面,这使客户端和服务端的动态信息交换成为可能。

JSP (Java Server Pages) 是由 Sun Microsystem 公司推出的新技术,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 (tag), 就构成了 JSP 网页 (*.jsp)。Web 服务器在遇到访问 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 的格式返回给客户。程序片段可以操作数据库、可以重新定向网页、可以发送 E-mail 等, 这就是建立动态网站所需要的功能。所有程序操作都在服务器端执行, 网络上传送给客户端的仅是得到的结果, 对客户浏览器的要求很低, 可以实现 No Plugin、No ActiveX 和 No Java Applet, 甚至无框架。

JSP 使得页面的动态内容与它的表示进一步分离。JSP 的设计迎合了两个不同的技术层面: 页面的图形设计和创建动态内容的软件开发。JSP 的主要特性如下。

- ☐ JSP 是基于文本的文档, 用来处理客户端请求 (request) 和构造响应 (response)。
- ☐ 构造 server 端访问的对象。
- ☐ 定义扩展 JSP 语言的机制。

JSP 与 Microsoft 的 ASP 技术非常相似。ASP 的编程语言是 VBScript 之类的脚本语言, JSP 使用的是 Java, 这是两者最明显的区别。此外, ASP 与 JSP 还有一个更为本质的区别: 两种语言引擎用完全不同的方式处理页面中嵌入的程序代码。在 ASP 下, VBScript 代码被 ASP 引擎解释执行; 在 JSP 下, 代码被编译成 Servlet 并由 Java 虚拟机执行, 这种编译操作仅在对 JSP 页面的第一次请求时发生。

JSP 在跨平台、执行速度等特性上具有很大的技术优势, 主要体现在以下方面。

- ☐ 将内容的生成和显示进行分离。
- ☐ 生成可重用的组件。
- ☐ 采用标识简化页面。
- ☐ 健壮的存储管理和安全性。
- ☐ 一次编写, 随处运行。

1.1.3 Ajax 概述

2005 年 2 月, Adaptive Path 的 Jesse James Garrett 最早创造了 Ajax 这个词。Ajax 将改变用户在网站上的体验, 之前的单击显示页面式的体验将变成具有交互性、灵活性的桌面应用程序体验。用户可以避免页面重新加载, 避免等待下一部分内容的呈现。

Ajax 的出现给 JavaScript 带来了生机, 应用和测试框架再加上更优秀的工具支持, 减轻了开发人员的重担。Ajax 在大多数浏览器中都能够使用, 并且不需要任何专门的软件或硬件。这种方式的一大优势就是开发人员不需要学习一种新的语言, 也不必丢弃原先掌握的服务器端技术。Ajax 是一种客户端方法, 可以与其他任何服务器交互, 它不关心为其提供服务的服务器是什么。

因特网默认的请求/响应模式有了重大改变, 这是 Ajax 的核心所在, 尽管这并非全新的内容。Web 应用开发人员现在可以自由地与服务器进行异步交互, 完成许多原本只能在富客户端上完成的任务。一句话, Ajax 给用户带来了新的网络体验。

Ajax 使浏览器可以为用户提供更为自然的浏览体验。在 Ajax 之前, Web 站点强制用户进入提交/等待/重新显示范例, 用户的动作总是与服务器的“思考时间”同步。Ajax 提供与服务器异步通信的能力, 从而使用户从请求/响应的循环中解脱出来。

借助于 Ajax, 可以在用户单击按钮时, 使用 JavaScript 和 DHTML 立即更新 UI, 并向服务器发出异步请求, 以执行更新或查询数据库。当请求返回时, 就可以通过 JavaScript 来改变 DOM, CSS 来相应地更新 UI, 而不是刷新整个页面。最重要的是, 用户甚至不知道浏览器正在与服务器通信: Web 站点看起来是即时响应的。虽然 Ajax 所需的基础架构已经出现了一段时间, 但直到最近异步请求的真正威力才得到利用。能够拥有一个响应极其灵敏的 Web 站点确实激动人心, 因为它最终允许开发人员和设计人员使用标准的 HTML/CSS/JavaScript 堆栈创建“桌面风格 (desktop-like)”的可用性。

Ajax 的核心是 JavaScript 对象 XMLHttpRequest。该对象在 Internet Explorer 5 中首次引入, 它是一种支持异步请求的技术。简而言之, 通过 XMLHttpRequest 对象, JavaScript 可以向服务器提出请求并处理响应, 而不阻塞用户。Ajax 与服务器交互的过程如图 1-1 所示。

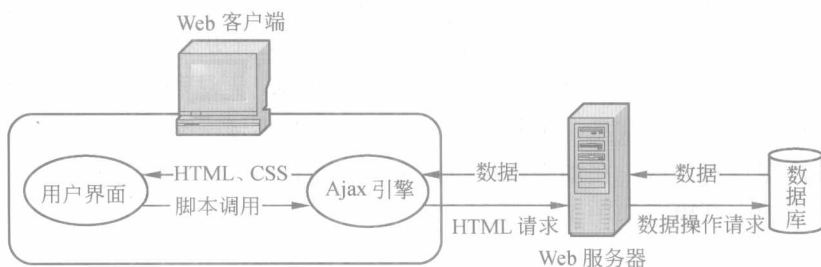


图 1-1 Ajax Web 应用程序模型

当然 Ajax 并不完美, 它可能破坏浏览器【后退】按钮的正常行为。在动态更新页面的情况下, 用户无法回到前一个页面状态, 因为浏览器仅能记忆历史记录中的静态页面。一个被完整读入的页面与一个已经被动态修改过的页面之间的差别非常微妙; 用户通常会希望单击【后退】按钮能够取消他们的前一次操作, 但是在 Ajax 应用程序中, 这将无法实现。开发者们想出了种种办法来解决这个问题, 大多数都是在用户单击【后退】按钮访问历史记录时, 通过创建或使用一个隐藏的 IFRAME 来重现页面上的变更 (例如, 当用户在 Google Maps 中单击【后退】按钮时, 它在一个隐藏的 IFRAME 中进行搜索, 然后将搜索结果反映到 Ajax 元素上, 以便将应用程序状态恢复到当时的状态)。

1.2 JSP+Ajax 结构及其意义

JSP 是跨平台性非常好的服务器端技术, 具有“一次编写, 到处运行”的特点。Ajax, 即异步 JavaScript 与 XML, 是 Web 2.0 中的一项关键技术, 它允许用户和 Web 页面间的交互与 Web 浏览器和服务端间的通信分离开来。将 JSP 和 Ajax 整合, 可以得出一种新的编程模式。

1.2.1 Web 应用解决方案

目前在 Internet 上占优势的三层客户端/服务器模型, 是通过通用网关接口 CGI

(Common Gateway Interface) 实现的。CGI 与 Web 几乎同时出现，曾经是提供 Web 交互性的唯一方法，所有标准的 Web 服务器软件都支持 CGI。CGI 本身可以实现为一个执行程序，这种 Web 与应用系统的集成方法非常简单、直接，因为它不需要对 Web 服务器与浏览器作任何修改。在这种体系结构中，Web 服务器通过 CGI 接口创建网关进程，并将用户请求传递给网关。应用网关处理 CGI 请求，并通过应用接口访问具体应用，然后将结果转换为 HTML 格式并回送给服务器。然而，CGI 的基础 HTTP 是一个低速、脆弱和无状态的协议，所以在利用 CGI 进行应用集成时存在一些问题。

Web 服务器对每一个用户请求都必须创建一个单独的网关进程，并在接到通过进程管道发回的结果后终止该进程。可见，通过 CGI 执行程序连接应用系统的缺点是性能开销较大，其中包括进程创建与终止开销、进程间的数据复制开销以及进程间的切换开销等。如果一个 HTML 文件中包含了大量的数据库访问请求，那么这种 CGI Web 网关的性能将十分低下，且服务器很快会达到饱和状态，难以支持更多的用户请求。另外，CGI 也无法适应于广泛流行且迅速发展着的 Java 等新兴技术，无法利用这些技术所带来的巨大优越性。这种现象非常常见，更新一个网页的一个数据或一个图片，需要将整个网页信息从服务器端重新加载，在加载过程中，可能还要花费大量时间等待。

这种传统的交互模式如图 1-2 所示。



图 1-2 传统 Web 应用模式

因此有人说，当前的 Web 服务器还处在中世纪，而客户端已经进入了后现代时期了。某些 Web 服务器开发商试图以专有服务器 API 的形式扩展 CGI，如 Netscape 的 NSAPI 和 Microsoft 的 ISAPI、NeXT 的 WebObjects 框架以及 Oracle 的 WebServerAPI 等。这种封闭的倾向是一条死胡同，它导致了到处充满非标准服务器的 Web。

传统的 Web 应用解决方案，将所有的操作集成在服务器端进行，客户端所做的工作只是简单的显示网页。形成了现在所说的胖服务器端和瘦客户端。Ajax 技术的出现，改变了传统 Web 开发的模式。通过 Ajax 技术，可以将大量的服务器端工作移植到客户端完成。这种新的交互模式如图 1-3 所示。

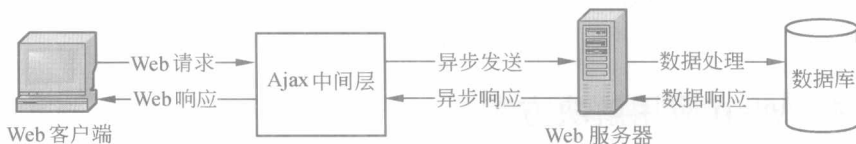


图 1-3 Web 2.0 应用模式

1.2.2 Ajax 解决方案的优势

当用户需要一种能够加快交互速度并且不需要在本地进行安装的 Web 技术时,市场为什么会在众多技术中选择了 Ajax 呢? 主要因为 Ajax 有以下优势。

□ 基于公开的标准

Ajax 技术是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,而不是那些热点的、最新的和未经考验的技术。现在,对于绝大多数的用户和企业来说,浏览器是一个可信任的应用平台。对于 Ajax 来说,Firefox 浏览器的基础 Mozilla 的发布是一个转折点,它开始支持 Ajax 技术。这种允许异步数据交换的技术早被 IE 浏览器支持了。这种支持和 Firefox 浏览器的大量被采用,真正地使人们理解了跨浏览器的富 Internet 应用成为了可能。

□ 高可用性

开发人员和设计人员不仅开始意识到大型的用户体验在市场上是成功的,而且也认识到这种体验是怎么来影响用户的开销的。Ajax 技术是使网络应用有更好的可用性的技术。它允许从服务器端请求少量的信息,而不是整个网页。它增加了页面数据的更新但同时减少了页面的刷新和刷新等待。人们需要一个优秀的用户界面,前提条件是用户能够快速取得信息不管数据是一个内部网的应用还是一个广域网的服务。

□ 跨平台跨浏览器的兼容性

IE 和基于 Mozilla 的 Firefox 是占据市场分额最大的两个浏览器,并且都支持在浏览器上轻松创建基于 Ajax 的 Web 应用。开发运行在先进的 Web 浏览器上的基于 Ajax 的富 Internet 应用成为可能,这是 Ajax 应用变得如此流行的一个最重要的原因。其实很多开发人员多年前就意识到 Ajax 技术流行的可能,但一直没有流行,是因为浏览器厂商的原因。

□ 高采用率

广泛采用 Ajax 技术已被业内领袖证明了市场的欢迎程度和该技术组的正确性。每一个该技术的使用者都成为了胜利者:包括 Google、yahoo、Amazon 和微软等。是 Google 地图吸引了 Web 开发人员的目光,人们由此才揭开了罩在 Ajax 头上的面纱。

□ 技术独立性

和 Ajax 技术的浏览器的独立性相同,该技术也兼容所有的标准型的服务器和服务端语言,如 PHP、ASP、ASP.Net、Perl、JSP 和 Cold Fusion 等。可以选择上述服务器端技术中的任意一个。这使得 Ajax 开发独立,因为所有开发人员都能使用并且一起讨论相同的表现层。

1.3 构建 JSP 运行环境

JSP 的运行环境包括两个部分,分别为 JDK 开发工具和 JSP 服务器。JDK 开发工具

包负责编译和解释执行 Java 文件，而 JSP 服务器负责将 JSP 文件转换为 Java 文件、提供事务和安全等方面的帮助。JSP 的服务器有很多类型，如 Tomcat、Resin、JBoss 和 WebLogic 等。

Tomcat 是 Jakarta 项目中的一个重要的子项目，又是 Sun 公司官方推荐的 Servlet 和 JSP 容器，也比较适合于初学者。JDK（Java Development Kit，Java 开发工具包）是一种用于构建在 Java 平台上发布的应用程序、Applet 和组件的开发环境。

1.3.1 安装 JDK 开发工具包

JDK 是 Sun Microsystems 针对 Java 开发人员提供的软件开发工具包。自从 Java 推出以来，JDK 已经成为使用最广泛的 Java SDK（Software development kit）。用户在编写 Java 程序时，必须用到类库和 Java 语言规范。JDK 中还包括完整的 JRE（Java Runtime Environment，Java 运行环境），也被称为 private runtime。包括了用于产品环境的各种库类，以及给开发员使用的补充库，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。

从初学者角度来看，采用 JDK 开发 Java 程序能够很快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。但它的缺点也非常明显，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。

Java 开发工具包（JDK）是 Java 程序员最初使用的开发环境。JDK 由一个标准类库和一组，测试及建立文档的 Java 实用程序组成。其核心 Java API 是一些预定义类库，开发人员需要用这些类来访问 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。

1. 获取 JDK 开发工具包

可以从 Sun 的官方网站 <http://java.sun.com> 上下载最新版本 JDK，进入到 Java SE 6.0 的下载页面，如图 1-4 所示。

在下载窗口中，单击 Download 按钮，就可以下载了。这里下载的是 JDK 6 Update 7。下载完毕后，会发现一个名称为 jdk-6u7-windows-i586-p.exe 可执行文件。

2. 安装 JDK

找到下载的 JDK 文件 jdk-6u7-windows-i586-p.exe，就可以开始安装了，具体步骤如下。

（1）双击 jdk-6u7-windows-i586-p.exe 文件开始进行安装，将打开【许可协议】窗口，单击【接受】按钮，打开【自定义安装】窗口。

（2）在这个窗口，可以更改文件的安装路径以及是否安装某些组件。这里把它安装