



and print, especially for the popular. Has called mixed media
multimedia.
Others used to modify another

multimedia
multimedia
And [multimedia-mixing], [mix]
Out [multimedia-mixing], [mix]
And [multimedia-mixing], [mix]
Produced with a single verb
The combined use of several
media, music, and lighting, as
education or entertainment.

a multimedia presentation; a
campaign.

multimedia
multimedia
And [multimedia-mixing], [mix]
Out [multimedia-mixing], [mix]
And [multimedia-mixing], [mix]
Produced with a single verb

The use of several
and print, especially
popular. Has called mixed media

a multimedia presentation;
a campaign.

a multimedia presentation;
a campaign.

multimedia
multimedia
And [multimedia-mixing], [mix]
Out [multimedia-mixing], [mix]
And [multimedia-mixing], [mix]
Produced with a single verb
The combined use of

卢菊平 郭江杰
飞思科技产品研发中心

编著
监制

精通

JSP+XML+CSS

网络开发混合编程

CONVERGE EXPERT INTELLIGENCE TO
CREATE BIBLE OF WEB DEVELOPMENT



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>



卢菊平 郭江杰
飞思科技产品研发中心

编著
监制

精通

JSP+XML+CSS

网络开发混合编程

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内容简介

本书面向当前网络开发黄金架构 JSP+XML+CSS, 整合 Java 开发中最流行的编程语言、IDE 工具、WebServices, 全面剖析 Java 开发核心架构、数据描述与传输技术、数据表现技术的关键。本书以循序渐进、由浅入深的讲解方式, 由各具特色的不同深度的案例表达每一个关键的技术要点, 同时给出多个综合的大型应用案例, 全部案例源自作者多年实际 Web 开发的项目积累。

书中实例源代码请到 <http://www.fecit.com.cn> 的“下载专区”下载。

本书非常适合于希望掌握 Web 开发综合编程技术的初中级读者研读, 对各类 Web 编程爱好者和企业 Web 开发人员具有针对性的指导意义。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有, 侵权必究。

图书在版编目(CIP)数据

精通 JSP+XML+CSS 网络开发混合编程 / 卢菊平, 郭江杰编著. —北京: 电子工业出版社, 2006.5

(网站开发专家)

ISBN 7-121-02400-4

I. 精... II. ①卢...②郭... III. ①JAVA 语言—主页制作—程序设计②可扩展语言, XML—程序设计③主页制作—软件工具, CSS IV. ①TP393.092②TP312

中国版本图书馆 CIP 数据核字 (2006) 第 021998 号

责任编辑: 赵红梅

印刷: 北京天宇星印刷厂

出版发行: 电子工业出版社

北京海淀区万寿路 173 信箱 邮编: 100036

经销: 各地新华书店

开本: 850×1168 1/16 印张: 28.5 字数: 775.2 千字

印次: 2006 年 5 月第 1 次印刷

印数: 7 000 册 定价: 45.80 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。质量投诉请发邮件至 zltts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

时代不断前进，而
技术之树常青。

——本书献给所有热爱 Web 开发的读者

Web 2.0 潮流下的技术困惑

现在，如果你不知道“Web 2.0”，不知道 Blog，不知道 RSS，就会遭到无数的白眼。因为这些词汇太火爆了，火爆的让普通的网民茫然，也让无数的开发者茫然。

这两年，在 IT 领域诞生了数不胜数的技术，商业的、开源的，每一次都给我们带来了新的冲击和新的诱惑。从 2005 年年底、2006 年年初引发的 Web 2.0 的潮流更是猛烈，Java 和 .NET 阵营趁势推出了众多产品和概念，让 Web 2.0 的浪潮一浪高过一浪。

现在的开发者真是累啊！既要掌握很多的技术，又要“与时俱进”。要想立足，要想有所发展，就要做到“人无我有，人有我精”。这个现象在 Web 开发领域更是突显，今天是 ASP，明天必须掌握 JSP，后天又要开始 .NET 的学习了。

彷徨之下，看来我们必须来深思一下技术的发展和应用的本质了。

Web 开发者的必备法宝

现在我们来画一画 Web 开发的“技术与应用拓扑”：



软件工程师可以根据需要进行组合！

不需要恐惧，我们先来看看作为一个网络软件工程师需要必备的法宝是什么，或者说你必须掌握什么技术体系才能够搭建自己的黄金开发平台。

随着网络应用的发展，系统越来越复杂，越来越庞大，而现在 Web 开发应用，不仅吸收了 C/S 应用的丰富特点，而且对现在的系统开发提出了更高的要求。

不要被概念炒作的迷雾所蒙蔽，不论 Web 2.0 这些最热的应用如何实现，其实仍然也是由最基本的技术进行重新组合后所开发出的新的应用，现在流行的 AJAX（Asynchronous Javascript and XML，

异步的 Javascript 和 XML) 技术也是如此, 在这个应用的背后不论发生多少变化, 基础的技术将是永恒的支持者。

所以, 对于广大的开发人员来说, 真正掌握这些知识并且利用它们才是最重要的。不论这些东西有多红火, 最基本的应用和中心还是要围绕着 XML 来进行, 它最美丽的外表是用 CSS 来化妆的, 它对后台数据的操作还需要 ASP、JSP、ASP.NET、PHP 等。

所以, 就技术而言, 左边显示的这个体系才是网络软件工程师的必备法宝, 缺一不可。



关于本书

但是, 真正介绍面向全能型开发技术架构的图书似乎不多, 所以我们组织了开发 Web 2.0 应用的第一线的资深程序员和项目经理编写了本书, 书中的实例从实战出发, 结合实际工作中的经验, 来介绍如何才能高效率地从 XML 文档中提取、过滤和合并数据, 并在应用程序中使用及显示它们。

不论你在学习 AJAX 还是其他的应用开发, 如果你从事 Web 开发, 你就不会跳过 XML 和 CSS 这个应用组合的门槛儿, 本书面向当前网络开发黄金架构 JSP+XML+CSS, 整合 Java 开发中最流行的编程语言、IDE 工具、WebServices, 全面剖析 Java 开发核心架构、数据描述与传输技术、数据表现技术的关键。本书以循序渐进、由浅入深的讲解方式, 由各具特色的不同深度的案例表达每一个关键的技术要点, 同时给出多个应用案例。全书全部案例源自作者多年实际 Web 开发的项目积累。

此书给出了整个开发环境的搭建, 从 JDK 和 Tomcat 的服务环境的配置, 再到目前流行的 Eclipse 工具和开发环境的搭建, 每一个实例都从最基本的环节开始, 一步一步带你进行开发。

需要说明的是, 在本书最后一章给出了一个整合的实例, 把全书前面的实例通过一个实例进行整合, 如果读者感兴趣, 可以看完本书前面的基础环境的搭建知识, 直接到第 11 章, 把系统的整个实例全部搭建起来, 当然, 你也可以从头开始学习, 辛勤耕耘, 直到最后一章去看最后的战果。

本书由飞思科技产品研发中心策划并组织编写, 由卢菊平、郭江杰主笔, 全书由邓健统稿。最后真心地希望此书能够给广大读者以帮助。

在本书的编写过程中, 我们力求精益求精, 但难免存在一些错误与不足, 敬请广大读者批评指正, 更欢迎大家与我们一起交流。

飞思科技产品研发中心

联系方式

咨询电话: (010) 68134545 88254160

电子邮件: support@fecit.com.cn

服务网址: <http://www.fecit.com.cn> <http://www.fecit.net>

通用网址: 计算机图书、飞思、飞思教育、飞思科技、FECIT

目 录

第 1 章 JSP+XML+CSS 模式开发	1
1.1 JSP 概述	1
1.1.1 B/S 系统结构	1
1.1.2 动态网页技术	2
1.1.3 JSP 技术	3
1.1.4 JSP 应用	4
1.1.5 JSP 程序初探	5
1.2 XML 概述	6
1.2.1 XML 的起源	7
1.2.2 XML 基础	8
1.2.3 XML 应用	10
1.2.4 XML 初探	12
1.3 CSS 概述	16
1.3.1 样式单——CSS	16
1.3.2 CSS 显示 XML 实例	17
1.4 JSP+XML+CSS 技术联合模式	19
1.4.1 基于 XML 的 Web 体系结构	19
1.4.2 JSP、XML 与 CSS 技术的融合	21
1.5 常用 JSP 服务器	24
1.5.1 Apache 服务器	24
1.5.2 Tomcat 服务器	24
1.5.3 BEA WebLogic	24
1.5.4 IBM WebSphere	25
1.5.5 JBoss 服务器	25
1.6 Java/JSP/XML 开发工具	26
1.6.1 文本编辑器	26
1.6.2 Web 开发工具	27
1.6.3 集成开发环境	27
1.7 本章小结	30
第 2 章 JSP 中生成动态 XML 实例	31
2.1 JDK 的安装与配置	31
2.1.1 J2SDK 1.5.0 的下载与安装	31
2.1.2 配置 J2SDK 1.5.0	33
2.1.3 测试 J2SDK	33
2.2 Tomcat 的安装与配置	34
2.2.1 Tomcat 的下载与安装	34
2.2.2 Tomcat 的配置	36
2.2.3 Tomcat 的测试	36
2.3 Eclipse 开发工具	36
2.3.1 Eclipse 的安装与配置	37
2.3.2 Tomcat 插件的安装与配置	38
2.3.3 JSP 插件的安装与配置	39

2.3.4	XML 插件的安装与配置.....	40
2.4	JSP 中直接生成 XML、CSS 显示实例.....	40
2.4.1	JSP 文件.....	44
2.4.2	CSS 文件.....	46
2.4.3	ANT 插件部署.....	48
2.4.4	错误提示页面.....	51
2.5	使用 JavaBean 生成动态 XML 实例.....	52
2.5.1	JSP 文件.....	52
2.5.2	Java 文件.....	56
2.5.3	XSL 文件.....	58
2.5.4	显示效果.....	63
2.5.5	提高与思考.....	63
2.6	本章小结.....	63
第 3 章	JSP 中用 XML 实现弹出式菜单.....	65
3.1	实例介绍.....	65
3.1.1	弹出式菜单的 JSP+XML 实现.....	65
3.1.2	设计思路.....	66
3.1.3	新建实例工程.....	67
3.2	Java 重要类库——java.io.*.....	70
3.2.1	java.io 包的类层次结构.....	70
3.2.2	java.io 包中几个重要的类.....	71
3.3	JavaScript 概述.....	72
3.3.1	Script 和 JavaScript.....	72
3.3.2	JavaScript 的用处.....	73
3.3.3	JavaScript 的对象层次图.....	73
3.4	开发详解.....	74
3.4.1	XML 格式的 JSP 文件.....	74
3.4.2	JavaBean 类.....	79
3.4.3	CSS 和 XSL 显示.....	90
3.4.4	JavaScript 的应用.....	93
3.5	进阶与提高.....	94
3.5.1	实例完善.....	94
3.5.2	实例拓展.....	96
3.6	本章小结.....	97
第 4 章	在 JSP 中应用 XML DOM 实例.....	99
4.1	XML DOM 概述.....	99
4.1.1	XML DOM 的由来.....	99
4.1.2	XML DOM 简介.....	100
4.1.3	使用 XML DOM 的优点.....	100
4.1.4	DOM 组成.....	101
4.1.5	DOM 版本.....	101
4.2	DOM 编程.....	102
4.2.1	DOM 编程思想.....	102
4.2.2	DOM 实施方案.....	103
4.2.3	DOM 常用接口.....	103

4.2.4	JAXP 实现.....	106
4.3	实例介绍：迷你论坛.....	107
4.3.1	实例简介.....	107
4.3.2	实现方案.....	107
4.3.3	新建实例工程.....	110
4.4	开发详解.....	114
4.4.1	XML 文档.....	114
4.4.2	样式单——my.css.....	115
4.4.3	浏览页面——elook.jsp.....	116
4.4.4	JavaBean 类.....	120
4.4.5	登录功能.....	137
4.4.6	管理模块.....	142
4.4.7	发帖功能.....	146
4.5	进阶与提高.....	149
4.5.1	实例完善.....	149
4.5.2	实例扩展.....	151
4.6	本章小结.....	151
第 5 章	在 JSP 中应用 SAX 解析 XML 实例.....	153
5.1	SAX 概述.....	153
5.1.1	SAX 的由来.....	153
5.1.2	SAX 的优缺点.....	154
5.1.3	SAX 版本.....	154
5.1.4	SAX 的应用.....	155
5.2	SAX 编程.....	155
5.2.1	SAX 编程思想.....	155
5.2.2	SAX 实施方案.....	156
5.2.3	SAX 常用接口和类.....	157
5.3	实例介绍.....	161
5.3.1	实例简介.....	161
5.3.2	实现方案.....	161
5.3.3	新建实例工程.....	163
5.4	开发详解.....	164
5.4.1	XML 文档.....	164
5.4.2	JavaBean 类.....	165
5.4.3	CSS 样式单.....	180
5.4.4	JSP 页面.....	182
5.5	进阶与提高.....	190
5.6	本章小结.....	191
第 6 章	Tag Library 在 JSP 中的应用实例.....	193
6.1	JSP 应用 Tag Library 模式.....	193
6.1.1	自定义标签的功能.....	193
6.1.2	自定义标签特性.....	194
6.1.3	标签的基本组成.....	194
6.1.4	自定义标签的分类.....	195
6.2	实例介绍.....	195

6.2.1	实例简介	195
6.2.2	实现方案	196
6.2.3	新建实例工程	196
6.3	开发详解	199
6.3.1	显示页面标题的处理器	199
6.3.2	格式化输出当天日期处理器	201
6.3.3	建立数据库连接处理器	202
6.3.4	获得数据库连接信息处理器	205
6.3.5	执行查询语句处理器	207
6.3.6	标签库描述文件	210
6.3.7	配置 web.xml 文件	213
6.3.8	在 JSP 中使用标签	213
6.3.9	CSS 样式文件	219
6.3.10	XSL 样式单文件	222
6.4	典型 Tag Library 的应用	223
6.4.1	JSTL 标记库简介	224
6.4.2	表达式语言介绍	225
6.4.3	JSTL 安装	226
6.4.4	JSTL 核心标签库简介	228
6.4.5	JSTL XML 标签库简介	228
6.4.6	JSTL formatting 标签库简介	228
6.4.7	数据访问标签库简介	228
6.5	进阶与提高	228
6.6	本章小结	229
第 7 章	JSP 联合 XML 与数据库实例	231
7.1	Microsoft SQL Server 2000 数据库	231
7.1.1	数据库介绍	231
7.1.2	SQL Server 2000 介绍	232
7.1.3	SQLServer 2000 的安装	233
7.1.4	SQL Server 2000 常用工具简介	238
7.2	JDOM 介绍	238
7.2.1	JDOM 概述	238
7.2.2	JDOM 和 DOM	239
7.2.3	获得并安装 JDOM	240
7.2.4	JDOM 的用处	240
7.2.5	JDOM 主要包结构介绍	240
7.2.6	相关资源	241
7.3	实例介绍	241
7.3.1	实例简介	241
7.3.2	实现方案	242
7.3.3	新建实例工程	242
7.4	开发详解	244
7.4.1	数据库连接类	244
7.4.2	创建 XML 文件类	247
7.4.3	XML 数据插入数据库类	249
7.4.4	相关 JSP 文件	250

7.5	部署与测试.....	257
7.5.1	安装 SQL Server 2000 的 JDBC 驱动.....	257
7.5.2	通过 ODBC 访问 SQL Server 2000 数据库.....	258
7.5.3	程序调试常见问题简述.....	259
7.6	进阶与提高.....	260
7.6.1	实例完善.....	260
7.6.2	实例拓展.....	265
7.7	本章小结.....	266
第 8 章	XML 数据岛实例——图书馆在线查询.....	267
8.1	数据岛介绍.....	267
8.2	数据岛的处理方法.....	269
8.3	实例介绍：实例名称.....	270
8.3.1	实例简介.....	270
8.3.2	实现方案.....	271
8.3.3	新建实例工程.....	271
8.4	开发详解.....	273
8.4.1	产生 XML 文件类.....	273
8.4.2	页面文件.....	276
8.4.3	XML 文件.....	281
8.4.4	JavaScript 文件.....	283
8.5	进阶与提高.....	293
8.6	本章小结.....	299
第 9 章	XML 技术在 MVC 体系中的应用：个人求职管理系统.....	301
9.1	JSP 开发模式.....	301
9.1.1	软件模式.....	301
9.1.2	直接使用 JSP.....	301
9.1.3	JSP 与 JavaBean 模式.....	302
9.1.4	MVC 模式.....	302
9.2	控制层介绍.....	303
9.2.1	JSP 页面控制.....	304
9.2.2	控制器.....	305
9.2.3	Servlet 控制.....	305
9.2.4	MVC 结构中应用 XML 技术.....	308
9.3	实例介绍.....	309
9.3.1	实例简介.....	309
9.3.2	实现方案.....	309
9.3.3	新建实例工程.....	311
9.3.4	安装与部署.....	312
9.4	开发详解.....	312
9.4.1	公共文件.....	312
9.4.2	Servlet 初探——系统登录的实现.....	318
9.4.3	数据访问.....	323
9.4.4	XML 和 Servlet 技术实现应用程序工作流的驱动.....	337
9.4.5	用户管理模块.....	346
9.4.6	其他模块.....	351

9.5 进阶与提高.....	352
9.6 本章小结.....	353
第 10 章 XML 技术在 Struts 架构中的应用：文章发布系统	355
10.1 系统分析与设计.....	355
10.1.1 需求分析	355
10.1.2 模块设计	357
10.1.3 系统功能的扩展	357
10.2 系统架构分析.....	358
10.2.1 Struts 与 MVC.....	358
10.2.2 Struts 在本实例中的应用	359
10.2.3 Web 服务器.....	360
10.3 XML 数据源.....	361
10.3.1 XML 与数据库	361
10.3.2 存放文章的 XML 文档： articles.xml	361
10.3.3 存放文章的 XML 文档： groups.xml.....	362
10.4 网站总体设计.....	362
10.4.1 新建工程	362
10.4.2 页面风格	363
10.4.3 公共类和函数	365
10.4.4 Struts 配置.....	368
10.5 Struts 组件扩展	374
10.5.1 BaseModel.java	374
10.5.2 BaseForm.java	375
10.5.3 BaseAction.....	376
10.6 文章管理模块.....	377
10.6.1 首页介绍	377
10.6.2 Struts 组件开发.....	383
10.6.3 使用自定义标签技术实现分页	400
10.6.4 文章查看	405
10.6.5 文章发布	406
10.6.6 文章删除	410
10.6.7 全文检索	410
10.7 文章类别管理模块.....	412
10.8 本章小结.....	415
第 11 章 应用 JSP+XML+CSS 技术构建本书实例演示系统	417
11.1 实例介绍	417
11.1.1 实例简介	417
11.1.2 新建实例工程	418
11.2 开发详解.....	419
11.2.1 页面框架	419
11.2.2 树型菜单设计	424
11.2.3 自定义标签技术在本系统中的应用	437
11.3 部署与测试.....	444
11.3.1 实例部署	444
11.3.2 实例测试	445

第 1 章 JSP+XML+CSS 模式开发

XML (eXtensible Markup Language——可扩展标记语言) 是一种扩展的源标记语言, 是可以定义其他语言的语言。它克服了 HTML 的种种不足, 将网络上传输的文档规范化, 它可建立任何复杂层次的数据模型, 并作为一种网络系统中通用的数据交换格式。JSP 则提供了一系列的功能, 可以非常理想地和 XML 协同工作, 由于 JSP 是 Java 软件环境的一部分, 这使得它能充分利用 Java 的强大功能和很好的灵活性。而 CSS 则可以很好地应用于描述网页和 XML 文档的表现。基于三者各自的特点和优势, 使用 JSP+XML+CSS 模式开发在网络编程中得到了越来越广泛的应用。

在理解 JSP+XML+CSS 模式开发之前, 有必要先来了解 JSP、XML 和 CSS 各自的特点和应用。

1.1 JSP 概述

1.1.1 B/S 系统结构

在传统的客户机/服务器 (Client/Server, 简称 C/S) 体系结构的应用系统中, 数据存放在数据库服务器里, 用户通过客户机对数据库进行操作。这种为客户安装前端应用程序的方法已经无法使用 Internet 网络的特性, Internet 网上运行者的客户机分布较广, 工作的环境也千差万别。

基于浏览器/客户端 (Browser/Server, 简称 B/S) 的应用程序比 C/S 有着难以比拟的优势。B/S 系统由浏览器和服务器组成, 服务器包括 Web 服务器、数据库服务器、应用服务器、中间件等。数据和程序放在服务器端, 服务器可以有多层结构, 服务器执行必要的计算, 负责与数据库的交互工作, 并将结构发送给客户。这种模式改变了 C/S 模式中客户机与服务器之间千丝万缕的联系, 使得程序员集中精力开发服务器端的应用软件。

采用 B/S 模型开发应用程序有以下的优越性。

- (1) 几乎没有限制的客户端访问。
- (2) 开发模块化。
- (3) 部署和管理极其简化、维护方便。要更新一个应用程序只需要更改一个基于服务器的程序, 而不是像 C/S 模式那样要更新成千上万的安装在客户端的应用程序。
- (4) 网络负载轻。系统中所有复杂的数据计算操作都放在服务器端实现, 浏览器和服务器之间传送的数据仅仅是计算的条件和结构, 因此大大地奖励了网络通信量, 减轻了网络负载。
- (5) B/S 结构下软件一般只有初期一次性投入成本; 而 C/S 结构下软件则随着应用范围的扩大, 要求不断进行资本的投入。比如需要购买更为高级的服务器或者增加相应的管理人员等。

正是上述优越性, 软件工业正迅速地向着基于浏览器/客户端的多层次应用程序迈进。

这些快速增长基于 Web 的应用程序要求开发技术上的改进。静态 HTML 只能显示相对静态的内容, 因此有必要创建交互的基于 Web 的应用程序, 页面的内容是基于用户的请求或者系统的状态。JSP 是企业版 Java 2 平台 (J2EE) 的重要组成部分, 是一种基于环境的全新的服务器端的 Java 动态网页技术, 为企业级 Web 站点动态网页开发人员提供了一种非常简单却相当有效的创作方法。JSP 技术是对 Servlet 技术的一种扩展, 它为基于 Java 环境开发多层结构的动态 Web 应用程序提供了一种方便、快捷的方法。

要想了解 JSP 技术原理、JSP 的优势和应用,我们首先来了解动态网页技术。

○1.1.2 动态网页技术

我们知道 HTML 是编制网页的基本语言,但它只能用于静态的网页。当今的 Web 已经不是早期的静态信息发布平台,它已被赋以更丰富的内涵。现在,我们不仅需要 Web 提供所需的信息,还需要提供可个性化搜索的功能,可以收发 E-mail,可以进行网上销售,可以从事电子商务,等等。为实现以上功能必须使用更新的网络编程技术制作动态网页。所谓动态,指的是按照访问者的不同需要,对访问者输入的信息做出不同的响应,提供响应的信息。

动态网页技术的原理是:使用不同技术编写的动态页面保存在 Web 服务器内,当客户端用户向 Web 服务器发出访问动态页面的请求时,Web 服务器将根据用户所访问页面的后缀名确定该页面所使用的网络编程技术,然后把该页面提交给相应的解释引擎;解释引擎扫描整个页面找到特定的定界符,并执行位于定界符内的脚本代码以实现不同的功能,如访问数据库、发送电子邮件、执行算术或逻辑运算等,最后把执行结果返回 Web 服务器;最终,Web 服务器把解释引擎的执行结果连同页面上的 HTML 内容,以及各种客户端脚本一同传送到客户端。虽然,客户端用户所接收到的页面与传统页面并没有任何区别,但是,实际上页面内容已经经过了服务器端处理,完成了动态的个性化设置。下面我们简单介绍一下常用的几个动态网页技术。

1. CGI

早期使用的动态网页就是 CGI,即 Common Gateway Interface (通用网关接口)。可以使用不同的程序编写适合的 CGI 程序,如 Visual Basic、Delphi 或 C/C++等,通常将已经写好的程序放在 Web 服务器的计算机上运行,再将其运行结果通过 Web 服务器传输到客户端的浏览器上。我们通过 CGI 建立 Web 页面与脚本程序之间的联系,并且可以利用脚本程序来处理访问者输入的信息并据此做出响应。事实上,这样的编制方式比较困难而且效率低下,因为每一次修改程序都必须重新将 CGI 程序编译成可执行文件。最常用于编写 CGI 技术的语言是 PERL (Practical Extraction and Report Language, 文字分析报告语言),它具有强大的字符串处理能力,特别适合用于分割处理客户端 Form 提交的数据串,用它来编写的程序后缀为.pl。

CGI 技术已经发展得很成熟了,功能强大,例如新浪、网易、搜狐等网站的搜索引擎,用的就是 CGI 技术。

2. ASP

目前,人们普遍采用 ASP 技术作为动态网站建设的技术,虽然采用 ASP 开发服务器程序的技术已经很成熟,但是 ASP 技术也存在着很多缺陷,如安全性低、适应性差、维护复杂等。

Active Server Pages,即 ASP,是微软开发的一种类似 HTML (Hypertext Markup Language, 超文本置标语言)、Script (脚本)与 CGI (Common Gateway Interface, 通用网关接口)的结合体,它没有提供自己专门的编程语言,而是允许用户使用包括 VBScript、JavaScript 等在内的许多已有的脚本语言编写 ASP 的应用程序。ASP 的程序编制比 HTML 更方便且更有灵活性。它是在 Web 服务器端运行,运行后再将运行结果以 HTML 格式传送至客户端的浏览器。因此 ASP 与一般的脚本语言相比,要安全得多。

ASP 的优点:简单易学、安装使用方便、开发工具强大而多样化、人性化、效率高。

ASP 的缺点:局限于微软的操作系统平台之上。ASP 主要工作环境是微软的 IIS 应用程序结构,又因 Activex 对象具有平台特性,所以 ASP 技术不能很容易地实现在跨平台的 Web 服务器的工作。安全性、稳定性、跨平台性都会因为与 NT 的捆绑显现出来。它是一种 Script 语言,只能使用组件提高工作效率。无法实现企业级的功能:完全的集群、负载均衡。

3. PHP

即 Hypertext Preprocessor (超文本预处理器), 是一种 HTML 内嵌式的语言。它的语法混合了 C、Java、PERL, 以及 PHP 式的新语法。PHP 是完全免费的, 可以不受限制地获得源码。它与 HTML 有很好的兼容性, 可以直接在脚本代码中加入 HTML 标签, PHP 还提供了标准的数据库接口, 数据库连接方便, 兼容性强, 扩展性强, 可以进行面向对象编程。

PHP 的优点: 能快速学习, 跨平台, 有良好的数据库交互能力; 与 Apache 及其他扩展库结合紧密, 有良好的安全性。

PHP 的缺点: 数据库支持变化极大; 安装复杂; 缺失企业级的支持; 缺少正规的商业支持; 无法实现商品化应用的开发。

1.1.3 JSP 技术

可以看出, CGI、ASP 和 PHP 等动态网页技术, 都有着它们先天的不足。它们都无法解决业界所需要的创建动态网页内容不受以下限制。

- 能够在任何 Web 或应用程序服务器上运行;
- 将应用程序逻辑和页面显示分离;
- 能够快速地开发和测试;
- 简化开发基于 Web 的交互式应用程序的过程。

JSP 技术便是设计用来满足这些要求的, 可以说它大大加速了动态网页技术的发展。

JSP (Java Server Pages) 是由 Sun Microsystem 公司于 1999 年 6 月推出的新技术, 是基于 Java Servlet 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, 这是因为它具有以下特点。

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 (内容是根据请求来变化的, 例如请求账户信息或者特定的一瓶酒的价格)。生成内容的逻辑被封装在标识和 JavaBeans 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 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 (例如, 通过访问 JavaBeans 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。

2. 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 (JavaBeans 或者 Enterprise JavaBeans™ 组件) 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, 使得这些组件为更多的使用者或者客户团体所使用。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。

3. 采用标识简化页面开发

Web 页面开发人员不会都是熟悉脚本语言的编程人员。JavaServer Page 技术封装了许多功能, 这些功能是在易用的、与 JSP 相关的 XML 标识中进行动态内容生成所需要的。标准的 JSP 标识能够访问和实例化 JavaBeans 组件, 设置或者检索组件属性, 下载 Applet, 以及执行用其他方法更难于编码和耗时的功能。

4. JSP 的适应平台更广

这是 JSP 比 ASP 的优越之处。几乎所有平台都支持 Java, JSP+JavaBean 可以在所有平台下通行无阻。NT 下 IIS 通过一个插件, 例如 JRUN 或者 ServletExec, 就能支持 JSP。著名的 Web 服务器 Apache 已经能够支持 JSP。由于 Apache 广泛应用在 NT、UNIX 和 Linux 上, 因此 JSP 有更广泛的运行平台。虽然现在 NT 操作系统占了很大的市场份额, 但是在服务器方面 UNIX 的优势仍然很大, 而新崛起的 Linux 更是来势不小。从一个平台移植到另外一个平台, JSP 和 JavaBean 甚至不用重新编译, 因为 Java 字节码都是标准的, 与平台无关。

与硬件平台和服务器无关性。JSP 作为 Java 家族的一员, 秉承了 Java 技术的“一次编写, 到处运行”的特性, 可以运行于大多数流行的操作系统平台和 Web 服务器, 这种与服务器硬件和操作系统平台的无关性是 JSP 相对于其他动态网页技术最大的优点。

Java 中连接数据库的技术是 JDBC (Java Database Connectivity)。很多数据库系统带有 JDBC 驱动程序, Java 程序就通过 JDBC 驱动程序与数据库相连, 执行查询、提取数据等操作。Sun 公司还开发了 JDBC-ODBC Bridge, 用此技术 Java 程序就可以访问带有 ODBC 驱动程序的数据, 目前大多数数据库系统都带有 ODBC 驱动程序, 所以 Java 程序能访问诸如 Oracle、Sybase、MS SQL Server 和 MS Access 等数据库。

5. JSP 的适应平台更广

JSP 提供了一种隐含对象在 JSP 页面中可以直接引用, 而不必首先声明, 利用 JSP 提供的这些隐含对象, 可以使脚本功能更加强大, 并且编程更加容易、方便。

可以说 JSP 是未来发展的趋势。世界上一些大的电子商务解决方案提供商都采用 JSP/Servlet。

1.1.4 JSP 应用

JSP 页面由 JSP 引擎执行, 通常使用 Web 服务器或使用 JSP 的应用服务器, 它接受客户端对 JSP 页面的请求。如图 1-1 所示是 JSP 页面程序中客户端请求和响应的过程。

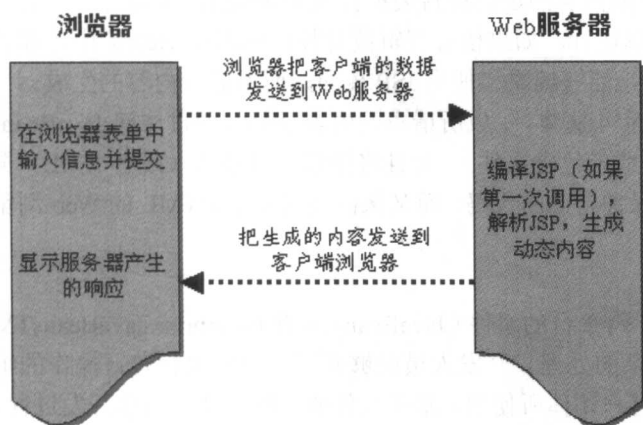


图 1-1 客户端请求和响应的过程

JSP 页面通常会编译成 Java Servlet, 这使得开发人员可以充分利用 Java 技术的扩展性和可移植性。JSP 页面只是在第一次调用时会被编译成一个 JSP Servlet 类, 存储在服务器的内存中。所以以后调用该页面时有很快的响应速度。

JSP 页面可以应用在不同的体系结构或模型中, 这里简单列举如下。

1. 简单应用

浏览器直接调用 JSP 页面, JSP 页面自己生成被请求的内容, 可以通过调用 JDBC 直接从数据库获取的信息。JSP 页面使用 JDBC 和 JavaBean 技术生成结果, 创建标准的 HTML 后将结果返回给浏览器。这个模型中 JSP 编译成为 Java Servlet, 它代替了 CGI-BIN 概念。它可以使得页面设计人员很容易根据请求和资源动态生成动态内容。在很多应用中采用了这个结构, 但它无法扩展到大量基于 Web 的并发客户访问稀少的企业资源, 它不可能为每个用户都建立或共享一个到可用内容资源的连接。举个最简单的例子, 比如访问数据库, 当出现并发访问时, 势必会影响到数据库的性能。

2. 与 Java Servlet 结合

在这种结构中, 基于 Web 的客户机可能直接请求 Java Servlet, Servlet 会生成动态内容, 将结果绑定到一个结果对象中并且调用 JSP 页面, JSP 页面从该对象中访问动态内容, 并且将结果发送给浏览器。

这个方法创建了更多的可以为应用程序间共享的可重用的组件, 并且可以作为更大的应用的一部分组成。但是它存在着和第一种模型扩展性差一样的问题。

3. 采用企业级 JavaBean 技术的可扩展处理

JSP 页面可以作为企业级 JavaBean (EJB) 体系结构中的一个中间层。在这种情况下, JSP 页面和后端资源通过 EJB 组件进行交互。关于 EJB 的概念读者可以参考相关的书本。EJB 组件管理对后端资源的访问, 从而为众多的并发使用者提供可扩展的性能。对于电子商务或其他应用, EJB 管理交易和潜在的安全性, 这将简化 JSP 页面。这个模型将为 Java 2 企业版 (J2EE) 所支持。

1.1.5 JSP 程序初探

前面介绍了这么多 JSP 知识, 读者可能会有点迫不及待了。这里就给大家演示一个比较简单的 JSP 程序——浏览器显示“你好 JSP”, 并且控制字体的大小变换。

运行本程序之前, 可以参考第 2 章安装 Tomcat 5.0 服务器, 我们这里只让大家对 JSP 有个感性的印象。

如图 1-2 所示是 JSP 程序的运行效果, 程序 1-1 所示是它对应的源代码。

你好 JSP
你好 JSP
你好 JSP
你好 JSP
你好 JSP
你好 JSP
你好 JSP
你好 JSP
你好 JSP

图 1-2 Ex1-1.jsp 运行效果

程序 1-1 chapter1/Ex1-1.jsp

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN//">
<%@ page language="java" contentType="text/html; charset=gb2312" %>
<%@ page info="a JSP example"%>
<html>
<body>
<center>
    <%
        //利用 For 循环控制字体由大到小
        for(int i=1; i<6; i++)
```

```

{
%>
<H<%=i%>>你好 JSP</H<%=i%>>
<%
}
//利用 For 循环控制字体由小到大
for(int i=5; i>0; i--)
{
%>
<H<%=i%>>你好 JSP</H<%=i%>>
<%
}
%>
</center>
</body>
</html>

```

程序的第一行是 JSP 页面中的习惯性说明, `language` 属性表明脚本语言的种类, 我们使用的是“Java”, 这也是目前 JSP 规范的默认值。`contentType` 和字符编码相关, `charset=gb2312` 代表使用 GB2312 进行编码, 这也是正确显示中文需要的编码方式。`info` 属性描述当前跟 JSP 文件的相关信息, 如页面信息。

`<%=i%>` 是 JSP 中获取变量值的语句, 其中里面的变量名可以是一个整型、字符型的变量。程序中使用的 For 语句语法是:

```

for(initialization Statement; conditional Expression; increment Statement)
{loopBody
}

```

为了执行 For 语句, 首先执行初始化语句, 然后再对条件表达式求值。如果值为 `true` 则执行循环体, 随后执行递增语句。表达式的求值和循环体与递增语句的执行不断重复, 直到条件表达式的值变为 `false`, 多个初始化和增量语句可以用逗号隔开。

For 循环的初始化部分可以声明局部循环变量, 循环变量的范围在循环体内部。这些循环变量遵循在块内有效的一般规则, 即用大括号作为程序块, 变量在块内声明, 不能与范围外边的变量同名, 也不能够在循环体外引用。

关于 JSP 中更多语法我们不再做过多的介绍了, 我们将会在后续的章节中陆续使用到。

注意: JSP 代码和 Java 一样, 对大小写是敏感的, 不可以有一点错误。一些类名、包名、路径名还有其他敏感的标签同样不能写错。

1.2 XML 概述

众所周知, 虽然 Internet 的发展不过十年, 但它对我们工作、学习、生活的方方面面却产生了深远的影响。一时间, 电子商务、网上出版、网络通信, 许许多多的新名词如潮水般涌来。在这个网络大潮之下, XML 这一以第二代网页发布语言而著称的新标准, 凭借着它的勃勃生机与强大优势, 为网络应用注入了新的活力。它被认为是继 Java 之后的又一颗网络编程的明星, 将给网络带来革命性的变化。