



Java Web开发详解

——XML+XSLT+Servlet+JSP

深入剖析与实例应用

孙鑫 编著

包含所有实例的代码、本书用到的软件和开发包的下载教学视频
**包含长达
30小时
的Java教学视频**



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>



Java Web开发详解

——XML+XSLT+Servlet+JSP

深入剖析与实例应用

孙鑫 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书共分4部分,从XML、Servlet、JSP和应用的角向读者展示了Java Web开发中各种技术的应用,循序渐进地引导读者快速掌握Java Web开发。

本书内容全面,涵盖了从事Java Web开发所应掌握的所有知识。在知识的讲解上,本书采用理论与实践相结合的方式,从程序运行的内部机制进行分析讲解,并通过大量的实例和实验来验证并运用本书的知识。

本书语言生动、通俗易懂、讲解细致,大部分章节都提供了多个例子,而且很多例子都是目前Web开发中经常使用的功能,具有相当的实用价值。

本书不仅可以作为Java Web开发的学习用书,还可以作为从事Java Web开发的程序员的参考用书和必备手册。

此外,本书的配套光盘还免费提供了价值人民币330元的Java教学视频,对Java语言进行了全面讲解,帮助一些不会Java语言的读者快速地从Java基础知识的学习中过渡到Java Web的学习与开发上。

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

Java Web开发详解:XML+XSLT+Servlet+JSP 深入剖析与实例应用 / 孙鑫编著. —北京:电子工业出版社, 2006.4

(Java技术大系)

ISBN 7-121-02396-2

I. J… II. 孙… III. JAVA语言—程序设计 IV. TP312

中国版本图书馆CIP数据核字(2006)第022000号

责任编辑:李冰 毕宁

印刷:北京天宇星印刷厂

出版发行:电子工业出版社

北京市海淀区万寿路173信箱 邮编100036

经销:各地新华书店

开本:850×1168 1/16 印张:55 字数:1610千字

印次:2006年4月第1次印刷

印数:6000册 定价:99.00元(含光盘1张)

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 zltz@phei.com.cn,盗版侵权举报请发邮件至 dbqq@phei.com.cn。

前言

随着 Java 语言的流行, Java 在网站和企业级应用的开发上应用得越来越普遍, Java Web 开发已经成为 Java 企业级解决方案中不可或缺的重要组成部分。

本书遵照 Servlet 2.4 和 JSP 2.0 规范, 系统、完整地介绍了 Java Web 开发中的各种技术, 从知识的讲解 → 知识的运用 → 实际问题的解决 → Web 开发框架 → 程序架构的设计, 一步一步地引导读者掌握 Java Web 开发的知识体系结构。

本书面向的读者

本书面向的读者群包括:

- 毫无 Web 经验的初学者。
- 有一定的 Web 经验, 但没有从事过 Web 开发的读者。
- 具有其他脚本语言 Web 开发经验, 想要快速转向 Java Web 开发的程序员。
- 正在从事 Java Web 开发的初中级程序员。

此外, 为了照顾到连 Java 语言都没有掌握的读者, 在本书的配套光盘中, 提供了一套完整的 Java 教学视频 (价值人民币 330 元), 以帮助读者快速地从 Java 语言的学习中过渡到 Java Web 的开发上。也就是说, 本书也适合于正准备学习 Java 语言的读者, 通过本书, 这些读者可以直接掌握 Java 语言, 到 Java Web 开发的企业级应用。

本书的内容组织

全书共分 4 部分, 包括了 XML 篇、Servlet 篇、JSP 篇和应用篇。前 3 部分相对独立, 但在内容上又有层进的关系。最后一部分综合应用了前 3 部分的知识, 讲解了一些高级应用, 并结合实际开发中要解决的问题, 给出了很多具有实用价值的实例程序。

此外, 附录作为本书不可或缺的部分, 对 HTML 和 HTTP 协议进行了介绍, 缺少 Web 经验的读者可以从这两章中学到 Web 开发的必备知识。另外, 附录还提供了 server.xml 和 web.xml 文件的介绍, 可以作为读者从事 Java Web 开发的参考。

本书在内容的编排上独具匠心, 将知识的连贯性和学习规律有机地组织在一起。本书首先介绍了目前 Web 开发中很热的 XML 技术, 通过和附录 A 的 HTML 知识进行对比学习, 读者可以很快地掌握 XML。在第 1 部分中, 还介绍了下面的内容:

- XML 名称空间
- 解析 XML 文档
- XSL 转换

本书第 2 部分是 Servlet 篇, 围绕 Servlet 讲解了 Servlet 技术和 Servlet 的应用。在这部分内容中, 还包括下面几个主题:

- Servlet 访问数据库
- 会话跟踪

- Servlet 的异常处理机制
- Servlet 的线程安全

读者如果掌握了这部分的内容,就可以开始 Web 应用程序的开发了。Servlet 容器和 Servlet 部署使用的配置文件都是 XML 格式,作者经常看到学员由于不了解 XML 语法而导致配置出错,从而影响程序的运行。也正是因为这种情况的普遍性,因此本书在第 1 部分就详细讲解了 XML,这样读者在配置 Web 容器和部署 Servlet 时,就不会因为语法问题而导致运行出错了。同时读者预先掌握了 XML 语法,对理解配置文件也有好处。

本书第 3 部分是 JSP 篇,包括了与 JSP 相关的各种技术。主要内容有 JSP 技术、JSP 与 JavaBean、JSP 开发的两种架构模型、自定义标签库、表达式语言、JSTL、标签文件。JSP 是建立在 Servlet 规范提供的功能之上的动态网页技术,读者看完第 2 部分介绍的 Servlet,自然也能理解 JSP 了。第 3 部分的内容非常详尽,不但可以作为学习使用,而且可以作为以后工作中的参考手册。

本书第 4 部分是应用篇,结合前 3 部分介绍的知识,讲解了一些高级应用,给出了很多具有实用价值的实例程序。主要内容有:

- 使用 Log4j 进行日志操作
- 过滤器在 Web 开发中的应用
- 中文乱码问题与国际化
- 使用 JBuilder2005 开发 Web 应用程序
- Web 应用程序安全
- 图像缩放、动态验证码、避免表单重复提交等实例
- 使用 Ant 辅助 Web 应用程序开发
- XML 与 XSLT 在 Web 开发中的应用

这部分内容从应用开发的角度帮助读者提升 Java Web 开发技能,建立 Web 开发框架设计的思想。

本书的实例程序

作者在编写本书时,使用的操作系统是 Windows 2000 Service Pack 4, Web 容器是 Tomcat 5.5.7, JDK 版本是 1.5.0_01,浏览器是 IE 6.0 SP1。本书所有的实例程序都在上述环境中运行正常。

本书绝大部分的例子程序,都没有放在 Tomcat 安装目录的 webapps 目录下,而是单独放在自定义的一个目录中。读者要运行这些程序,有两种方式:一种方式是直接将光盘中的例子程序目录复制到 Tomcat 的 webapps 目录下运行;另一种方式是配置 Web 应用程序的运行目录,在 %CATALINA_HOME%\conf\Catalina\localhost 目录下(%CATALINA_HOME%表示 Tomcat 的安装目录),建立 chX.xml (X 表示每章的序号,例如第 5 章是 ch05.xml,第 13 章是 ch13.xml)文件,编辑这个文件,输入下面的内容:

```
<Context path="/ch07" docBase="F:\JSPLesson\ch07" reloadable="true"/>
```

将 F:\JSPLesson\ch07 替换为读者机器上对应章节的 Web 应用程序所在的目录。

此外,每章例子程序的开发步骤中也给出了详细的配置过程,读者遵照步骤操作,会发现配置非常容易。

学习建议

作者针对三种不同类型的读者,提出下面的学习建议。

(1) 对 HTML 和 HTTP 协议不是很了解的读者：建议从附录 A、附录 B 开始学习，然后再从第 1 章开始学习。在第 1 章的学习中，要结合附录 A，彼此对照，看看 XML 和 HTML 之间的异同，加深学习的效果。在学习第 2 部分的 Servlet 时，要结合附录 B，随时参照，以加深对 HTTP Servlet 的理解。

(2) 对 HTML 和 HTTP 协议比较清楚的读者，可以直接从第 1 章开始学习，按照章节的顺序一步一步进行下去。

(3) 有 Java Web 开发经验的读者，可以选择性地学习本书的内容，并在开发过程中，以本书作为参考用书。

由于第 1 部分的第 4 章（XSL 转换）内容较多，而且有一定的难度，所以在这里建议读者在学习第 26 章（即最后一章）之前，再回过头学习这一章也可以。当然，对于能够理解这一章内容的读者，学习的次序也就不重要了。

本书的配套光盘说明

本书的配套光盘提供了 3 部分的内容：本书所有实例的代码、价值人民币 330 元的 Java 教学视频（Flash 格式）、本书用到的一些软件和开发包的下载教学视频。

1. 实例代码

配套光盘中的实例代码是按照章节的顺序提供的，例如，第 7 章的例子程序位于光盘的 JSPLesson\ch07 目录下。某些章节的例子程序运行需要配置依赖的软件或开发包，请参见各章的内容，书中用到的软件和开发包的下载可参看光盘中提供的下载教学视频。

2. Java 教学视频

本套教学视频由作者本人录制，市场定价为人民币 330 元。为了帮助广大 Java 爱好者能够快速地步入 Java 企业级的开发，作者花费了大量的时间和精力将 AVI 格式的视频转换为 Flash 格式，免费提供给读者使用，希望读者能够从中受益。

这套视频的内容简介如下。

Lesson1: Java 的一些基本概念，Java 的起源、我们为什么要学习 Java 语言，Java 跨平台原理剖析；Java 环境变量的配置，初学者常犯的问题，Java 语言的基本数据类型和 Java 的语句。

Lesson2: Java 面向对象编程，面向对象编程和面向过程编程的区别，我们如何才能掌握面向对象的编程，类和对象的关系；讲解了 Java 中的构造、重载、this 和 super 变量、静态变量、Java 中的常量、继承、多态。

Lesson3 上：Java 中包的概念，类的说明符、方法的说明符、对象的销毁（JVM 垃圾回收器的演示），Java 中接口理解与掌握。

Lesson3 下：Java 中内部类的理解、运用与掌握，Java 中的异常处理。

Lesson4 上：Java 的常用包，"=="和"equals"的用法，基本数据类型与引用类型，对象的克隆。

Lesson4 下：数组的复制与排序，Runtime 类与 Process 类，Class 类与反射 API，Runtime 类与单例设计模式。

Lesson5: Java 的多线程，程序、进程和线程的概念，实现多线程的两种方式，线程同步的原理，线程的死锁，运用 wait 和 notify 来实现 producer - consumer 关系，线程终止的两种情况。

Lesson6: Java 中的集合类的讲解，栈、队列、链表等数据结构的讲解。

Lesson7: Java I/O 操作，字节流 InputStream 和 OutputStream，字符流 Reader 和 Writer，Java I/O 库的设计原则，字符集的编码，RandomAccessFile 类，管道流，对象的序列化。

Lesson8: Java 图形界面编程, AWT、布局管理器、事件模型, JFC、Swing 编程。应用 JBuilder 快速开发图形界面程序。

Lesson9: Applet 编程, 详细讲解 Applet 的生命周期, 子体的设置, applet 从网页中获取信息, applet 和浏览器的通信, 更新显示的原理, 用 Applet 来制作动画程序, 以及动画程序的优化。

Lesson10: Java 网络编程, 基于 TCP 的套接字编程, 基于 UDP 的套接字编程, URL 和 URI, 一个实用的下载程序。

3. 下载教学视频

Java Web 开发中经常要用到某些开源软件或开发包, 有些初学者往往不知道这些软件或开发包在什么地方下载, 以及如何下载, 导致无法运行某些样例程序, 影响了学习效果。如果随盘提供这些软件, 又会牵涉到版权的问题, 而且由于图书发行时间的问题, 随盘提供的软件往往都比较旧, 为此, 作者专门录制了这段视频, 以帮助读者了解和掌握开源软件的下载方法。

进一步学习的建议

读者在掌握本书的内容后, 可以再花一到两个月的时间自学某种 Web 开发框架, 如 Struts (学完本书的内容, 再学 Struts 将是非常容易的), 以及某种对象-关系映射工具, 如 Hibernate。在掌握这些内容后, 读者就可以去公司从事 Java 企业级的开发了。很多人在谈到企业级开发时, 总是想到 EJB, 实际上, 很多公司即使在做税务、银行这样的系统, 也只是采用 Struts+Hibernate 这样的轻量型架构。

最后, 衷心地祝愿读者能够从此书获益, 从而实现自己的开发梦想。由于本书的内容较多、牵涉的技术较广, 错误和疏漏之处在所难免, 欢迎广大技术专家和读者指正。作者的联系方式是 csunxin@sina.com, 读者也可以上作者的网站发表意见, 网址是: <http://www.sunxin.org/>。

目 录

第 1 部分 XML 篇

第 1 章 XML 与 DTD	2
1.1 XML 的产生	2
1.2 W3C 介绍	2
1.3 关于 XML 的几个问题	3
1.4 XML 与 HTML 的比较	4
1.4.1 XML 将数据与显示分开	5
1.4.2 XML 对文档的格式要求更加严格	6
1.4.3 XML 有且只能有一个根元素	6
1.5 XML 的编辑工具	7
1.6 XML 文档	8
1.6.1 XML 声明	9
1.6.2 文档类型声明	10
1.6.3 元素	11
1.6.4 注释	15
1.6.5 处理指令	15
1.6.6 空白处理	16
1.6.7 行尾处理	16
1.6.8 语言标识	16
1.7 格式良好的 XML	17
1.8 DTD	18
1.8.1 在 XML 文档中引入 DTD	18
1.8.2 DTD 的结构	21
1.9 有效的 XML	35
1.10 XML 处理器 / 解析器	35
1.11 小结	36
第 2 章 XML 名称空间	38
2.1 声明名称空间	38
2.2 名称空间在元素和属性中的运用	39
2.2.1 名称空间在元素中的运用	39
2.2.2 默认名称空间	41
2.2.3 名称空间在属性中的运用	42
2.3 名称空间和 DTD	43
2.4 小结	45

第 3 章 对 XML 文档进行分析	46
3.1 DOM、SAX 和 JAXP	46
3.2 使用 DOM 解析 XML 文档	47
3.2.1 DOM 结构模型	47
3.2.2 DOM 解析器工厂	50
3.2.3 JAXP 的错误类和异常类	52
3.2.4 用 DOM 解析 XML 文档实例	53
3.3 使用 SAX 解析 XML 文档	65
3.3.1 SAX 的处理机制	66
3.3.2 配置 SAX 解析器	69
3.3.3 SAX 解析器工厂	70
3.3.4 SAX 的异常类	71
3.3.5 ErrorHandler 接口	73
3.3.6 使用 SAX 解析 XML 文档实例	74
3.4 JDOM	83
3.4.1 下载并配置 JDOM	83
3.4.2 JDOM API 介绍	84
3.5 dom4j	88
3.5.1 下载并配置 dom4j	88
3.5.2 dom4j API 介绍	88
3.5.3 第一个实例	92
3.5.4 第二个实例	94
3.6 解析名称空间	96
3.6.1 DOM 和名称空间	96
3.6.2 SAX 和名称空间	97
3.6.3 JDOM 和名称空间	98
3.6.4 dom4j 和名称空间	98
3.7 小结	99
第 4 章 XSL 转换	101
4.1 XSLT 概述	101
4.2 Xalan 处理器	105
4.3 模板规则	106
4.4 <xsl:apply-templates>元素	107
4.5 <xsl:value-of>元素	108
4.6 <xsl:for-each>元素	110
4.7 匹配节点的模式	111

4.8	mode 属性	113
4.9	内置的模板规则	114
4.10	对空白的处理	115
4.11	XPath 语言	116
4.11.1	XPath 上下文	116
4.11.2	位置路径	117
4.11.3	表达式	121
4.11.4	核心函数库	123
4.12	创建结果树	126
4.12.1	创建元素和属性	127
4.12.2	创建文本	131
4.12.3	创建处理指令	133
4.12.4	创建注释	133
4.12.5	复制节点	134
4.12.6	输出格式化的数字	135
4.13	条件处理	144
4.13.1	<xsl:if>	144
4.13.2	<xsl:choose>	145
4.14	排序	146
4.15	变量和参数	150
4.15.1	变量	150
4.15.2	参数	152
4.16	命名模板	153
4.17	合并样式表	154
4.17.1	导入样式表	154
4.17.2	包含样式表	155
4.18	模板规则冲突的解决	156
4.19	<xsl:output>元素	157
4.19.1	指定输出文档的格式	158
4.19.2	输出 XML 声明	158
4.19.3	输出文档类型定义	159
4.19.4	输出 CDATA 段	160
4.19.5	指定文档缩进	161
4.19.6	指定媒体类型	161
4.20	XSLT 中的函数	162
4.21	数字格式化	162
4.22	查询和分组	164
4.23	处理多个输入文档	172
4.24	JAXP 中的 XSLT API	175
4.24.1	转换器工厂	175
4.24.2	Transformer 和 Templates	176

4.24.3	一个例子	178
4.25	小结	179

第 2 部分 Servlet 篇

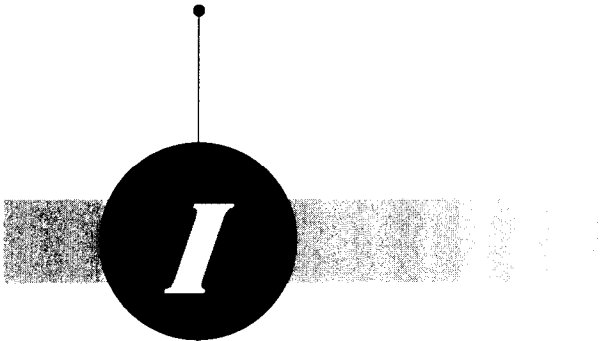
第 5 章	Servlet 与 Tomcat	182
5.1	Servlet 与 Servlet 容器	182
5.2	Servlet 容器的分类	183
5.3	Servlet 和其他技术的比较	183
5.4	Tomcat 介绍	183
5.5	Tomcat 的安装与配置	184
5.5.1	安装 Tomcat	185
5.5.2	运行 Tomcat	186
5.5.3	Tomcat 启动分析	189
5.5.4	Tomcat 的体系结构	191
5.6	Tomcat 的管理程序	193
5.6.1	admin Web 应用程序	193
5.6.2	manager Web 应用程序	194
5.7	小结	195
第 6 章	Servlet 技术	196
6.1	Servlet API	196
6.1.1	Servlet 接口	196
6.1.2	Servlet 相关接口与实现类	197
6.2	几个实例	205
6.2.1	实例一: SimpleHello	205
6.2.2	实例二: WelcomeYou	210
6.2.3	实例三: OutputInfo	216
6.2.4	实例四: LoginServlet	220
6.3	Servlet 异常	223
6.3.1	ServletException 类	224
6.3.2	UnavailableException 类	224
6.4	Servlet 生命周期	224
6.5	Servlet 上下文	226
6.5.1	ServletContext 接口	226
6.5.2	页面访问量统计实例	228
6.6	请求转发	231
6.6.1	RequestDispatcher 接口	232
6.6.2	得到 RequestDispatcher 对象	232
6.6.3	请求转发的实例	233
6.6.4	sendRedirect()和 forward()方法的 区别	238
6.7	小结	239

第 7 章 Web 应用程序的部署	240	10.1.1 HTTP 错误代码的处理	338
7.1 配置任意目录下的 Web 应用程序	240	10.1.2 Java 异常的处理	341
7.2 WAR 文件	242	10.2 程序式异常处理	343
7.3 Tomcat 中 Servlet 的另一种运行方式	244	10.2.1 在 try-catch 语句中处理异常	343
7.4 与 Servlet 配置相关的元素	247	10.2.2 使用 RequestDispatcher 来处理异常	346
7.4.1 <servlet>元素及其子元素	247	10.3 小结	349
7.4.2 <servlet-mapping>元素及其子元素	249	第 11 章 开发线程安全的 Servlet	350
7.5 一个实例	250	11.1 多线程的 Servlet 模型	350
7.6 小结	253	11.2 线程安全的 Servlet	351
第 8 章 数据库访问	254	11.2.1 变量的线程安全	351
8.1 驱动程序的类型	254	11.2.2 属性的线程安全	360
8.1.1 JDBC-ODBC 桥	254	11.3 SingleThreadModel 接口	362
8.1.2 部分本地 API Java 驱动程序	255	11.4 小结	363
8.1.3 JDBC 网络纯 Java 驱动程序	255	11.5 思考题	363
8.1.4 本地协议的纯 Java 驱动程序	256	第 3 部分 JSP 篇	
8.2 安装数据库	256	第 12 章 JSP 技术	366
8.3 JDBC API	260	12.1 JSP 简介	366
8.3.1 加载并注册数据库驱动	261	12.2 JSP 的运行机制	366
8.3.2 建立到数据库的连接	263	12.3 JSP 的语法	371
8.3.3 访问数据库	264	12.3.1 指令元素 (directive elements)	371
8.3.4 事务处理	289	12.3.2 脚本元素 (scripting elements)	374
8.3.5 可滚动和可更新的结果集	296	12.3.3 动作元素 (action elements)	376
8.4 JDBC 数据源和连接池	299	12.3.4 注释	383
8.5 MySQL 对中文的处理	302	12.4 JSP 的隐含对象	383
8.6 小结	302	12.4.1 pageContext	384
第 9 章 会话跟踪	303	12.4.2 out	385
9.1 用于会话跟踪的技术	303	12.4.3 page	385
9.1.1 SSL 会话	304	12.4.4 exception	386
9.1.2 Cookies	304	12.5 对象和范围	387
9.1.3 URL 重写	305	12.6 留言板程序	389
9.2 Java Servlet API 的会话跟踪	306	12.7 留言板管理程序	397
9.2.1 HttpSession 接口	306	12.8 JSP 文档	402
9.2.2 Session 的生命周期	307	12.8.1 JSP 文档的标识	404
9.2.3 Cookie 的应用	319	12.8.2 JSP 文档中的元素语法	404
9.2.4 HttpSessionBindingListener 接口	328	12.9 小结	407
9.2.5 在线人数统计程序	329	第 13 章 JSP 与 JavaBean	409
9.3 小结	335	13.1 JavaBean 简介	409
第 10 章 Servlet 的异常处理机制	337	13.1.1 属性的命名	409
10.1 声明式异常处理	338		

13.1.2 属性的类型	411	16.1.4 逻辑操作符	485
13.2 在 JSP 中使用 JavaBean	412	16.1.5 Empty 操作符	485
13.2.1 <jsp:useBean>	412	16.1.6 条件操作符	485
13.2.2 <jsp:setProperty>	413	16.1.7 圆括号	486
13.2.3 <jsp:getProperty>	414	16.1.8 操作符的优先级	486
13.2.4 示例	414	16.2 隐含对象	486
13.3 网上书店程序	418	16.3 命名变量	487
13.4 小结	440	16.4 保留的关键字	488
第 14 章 JSP 开发的两种模型	441	16.5 函数	488
14.1 模型 1	441	16.6 小结	489
14.2 模型 2	445	第 17 章 JSP 标准标签库 (JSTL)	490
14.3 小结	449	17.1 JSTL 简介	490
第 15 章 标签库 (Tag Library)	450	17.2 配置 JSTL	491
15.1 标签库 API	450	17.3 Core 标签库	491
15.1.1 标签的形式	451	17.3.1 一般用途的标签	492
15.1.2 Tag 接口	451	17.3.2 条件标签	496
15.1.3 IterationTag 接口	453	17.3.3 迭代标签	498
15.1.4 BodyTag 接口	454	17.3.4 URL 相关的标签	501
15.2 标签库描述符	455	17.4 I18N 标签库	506
15.2.1 <taglib>元素	456	17.4.1 国际化标签	506
15.2.2 <validator>元素	457	17.4.2 格式化标签	511
15.2.3 <listener>元素	457	17.5 SQL 标签库	520
15.2.4 <tag>元素	458	17.5.1 <sql:setDataSource>	520
15.2.5 <tag-file>元素	460	17.5.2 <sql:query>	521
15.2.6 <function>元素	461	17.5.3 <sql:param>	524
15.3 传统标签的开发	462	17.5.4 <sql:dateParam>	525
15.3.1 实例一: <hello>标签	462	17.5.5 <sql:update>	526
15.3.2 实例二: <max>标签	465	17.5.6 <sql:transaction>	527
15.3.3 实例三: <greet>标签	467	17.6 XML 标签库	529
15.3.4 实例四: <switch>标签	469	17.6.1 核心操作	529
15.3.5 实例五: <iterate>标签	473	17.6.2 流程控制	532
15.4 简单标签的开发	479	17.6.3 转换操作	536
15.4.1 SimpleTag 接口	479	17.7 Functions 标签库	539
15.4.2 实例一: <welcome>标签	480	17.7.1 fn:contains	539
15.4.3 实例二: <max_ex>标签	481	17.7.2 fn:containsIgnoreCase	539
15.5 小结	483	17.7.3 fn:startsWith	540
第 16 章 表达式语言 (EL)	484	17.7.4 fn:endsWith	541
16.1 语法	484	17.7.5 fn:indexOf	541
16.1.1 “[]” 和 “.” 操作符	484	17.7.6 fn:replace	542
16.1.2 算术操作符	484	17.7.7 fn:substring	543
16.1.3 关系操作符	485	17.7.8 fn:substringBefore	544

17.7.9	fn.substringAfter	544
17.7.10	fn.split	545
17.7.11	fn.join	546
17.7.12	fn.toLowerCase	547
17.7.13	fn.toUpperCase	547
17.7.14	fn.trim	548
17.7.15	fn.escapeXml	548
17.7.16	fn.length	549
17.8	小结	549
第 18 章 标签文件 (Tag Files)551		
18.1	标签文件的语法.....	551
18.2	一个简单的标签文件	551
18.3	标签文件的隐含对象	554
18.4	标签文件的指令.....	555
18.4.1	tag 指令.....	555
18.4.2	attribute 指令	556
18.4.3	variable 指令	557
18.5	标签文件实例讲解	558
18.5.1	实例一: <welcome>标签	558
18.5.2	实例二: <toHtml>标签	559
18.6	<jsp:invoke>动作元素	561
18.7	<jsp.doBody>动作元素	562
18.8	小结	562
第 4 部分 应用篇		
第 19 章 使用 Log4j 进行日志操作564		
19.1	Log4j 介绍.....	564
19.1.1	Logger 组件	564
19.1.2	Appender 组件	566
19.1.3	Layout 组件	567
19.2	使用 Log4j	568
19.3	Log4j 使用实例.....	572
19.4	NDC 和 MDC	585
19.5	小结	587
第 20 章 Filter 在 Web 开发中的应用.....588		
20.1	过滤器概述	588
20.2	Filter API	589
20.2.1	Filter 接口	589
20.2.2	FilterConfig 接口	590
20.2.3	FilterChain 接口.....	590
20.3	过滤器的部署	591
20.4	过滤器的开发	592
20.5	对用户进行统一验证的过滤器	594
20.6	对请求和响应数据进行替换的 过滤器	600
20.7	小结	609
第 21 章 中文乱码问题与国际化610		
21.1	中文乱码问题产生的由来.....	610
21.1.1	常用字符集	610
21.1.2	对乱码产生过程的分析	612
21.2	中文乱码问题的解决方案	614
21.3	使用过滤器解决中文问题	616
21.4	让 Tomcat 支持中文文件名	620
21.5	国际化与本地化	621
21.5.1	Locale	621
21.5.2	资源包	623
21.5.3	编写国际化的 Web 应用程序	625
21.6	小结	629
第 22 章 使用 JBuilder2005 开发 Web 应用程序.....630		
22.1	JBuilder2005 简介	630
22.2	集成开发环境介绍.....	631
22.2.1	菜单栏	631
22.2.2	主工具栏	632
22.2.3	工程工具栏和工程窗口	632
22.2.4	结构窗口	633
22.2.5	文件标签和内容窗口	634
22.2.6	文件视图标签	634
22.2.7	消息窗口	637
22.3	JBuilder2005 的基本配置	638
22.3.1	配置 JDK	638
22.3.2	配置 Web 服务器	640
22.4	文件的上传和下载	642
22.4.1	基于表单的文件上传	642
22.4.2	文件上传格式分析	643
22.4.3	commons-fileupload 组件	644
22.4.4	文件上传实例	646
22.4.5	文件下载实例	656
22.5	给图片添加水印和文字	664
22.6	小结	667

第 23 章 开发安全的 Web 应用程序	668	25.3.2 工程 (<project>)	712
23.1 概述	668	25.3.3 目标 (<target>)	713
23.2 理解验证机制	669	25.3.4 任务	714
23.2.1 HTTP Basic Authentication	669	25.3.5 Path-like Structures	728
23.2.2 HTTP Digest Authentication	670	25.3.6 引用 (References)	729
23.2.3 HTTPS Client Authentication	670	25.4 部署到 Tomcat	730
23.2.4 Form Based Authentication	671	25.5 运行 Ant	732
23.3 声明式安全	671	25.6 小结	732
23.3.1 <security-constraint>元素	671	第 26 章 XML 和 XSLT 在 Web 开发中的	
23.3.2 <login-config>元素	673	应用	733
23.3.3 基本验证的实现	674	26.1 功能概述	733
23.3.4 基于表单验证的实现	676	26.2 程序设计	733
23.3.5 使用数据库保存用户名和密码	678	26.2.1 软件架构	733
23.4 程序式安全	682	26.2.2 页面设计	735
23.5 SQL 注入攻击的防范	685	26.2.3 数据库设计	747
23.6 小结	687	26.3 程序实现	749
第 24 章 Web 应用程序开发实例	688	26.3.1 org.sunxin.guestbook	749
24.1 图像的缩放	688	26.3.2 org.sunxin.guestbook.beans	750
24.2 如何动态生成验证码	692	26.3.3 org.sunxin.guestbook.util	758
24.3 如何避免表单的重复提交	696	26.3.4 org.sunxin.guestbook.parser	762
24.3.1 在客户端避免表单的重复提交	697	26.3.5 org.sunxin.guestbook.controller	765
24.3.2 在服务器端避免表单的重复提交	700	26.3.6 org.sunxin.guestbook.view	787
24.4 小结	706	26.3.7 page.xml	791
第 25 章 使用 Ant 辅助 Web 应用程序		26.3.8 web.xml	797
开发	707	26.3.9 context.xml	798
25.1 Ant 介绍	707	26.4 小结	798
25.2 安装并配置 Ant	708	附录 A 快速掌握 HTML	799
25.2.1 下载并安装 Ant	708	附录 B 解析 HTTP	820
25.2.2 设置 Ant 的运行环境	708	附录 C server.xml 文件	833
25.3 编写 Ant 的构建文件	709	附录 D web.xml 文件	840
25.3.1 准备工作	709	索引	852



I

XML 篇

XML 已经成为大家耳熟能详的一个词汇，其应用也越来越多，但很多人对 XML 的知识只是一知半解，对 XML 领域的其他技术也知之甚少。为了帮助读者了解和掌握 XML 的相关技术，并为后续部分的学习奠定基础，我们在第一部分就为读者详细讲解 XML 技术领域中的应用最为广泛的 3 项技术（标准）：XML、XML 名称空间和 XSLT，并介绍它们和 Java 程序的结合与应用。

- ◇ 第 1 章 XML 与 DTD
- ◇ 第 2 章 XML 名称空间
- ◇ 第 3 章 对 XML 文档进行分析
- ◇ 第 4 章 XSL 转换

XML 与 DTD

XML 对我们来说已不再陌生，其相关概念和知识在网络上随处可见，有关 XML 的应用也越来越多，本章的目的是帮助读者快速了解和掌握 XML，为后面章节的学习打下基础。

1.1 XML 的产生

XML 的全称是 Extensible Markup Language，意思是可扩展的标记语言，它是标准通用标记语言（Standard Generalized Markup Language，SGML）的一个子集。那 SGML 又是什么呢？

在 20 世纪 80 年代早期，IBM 提出在各文档之间共享一些相似的属性，例如字体大小和版面。IBM 设计了一种文档系统，通过在文档中添加标记，来标识文档中的各种元素，IBM 把这种标识语言称做通用标记语言（Generalized Markup Language，GML）。经过若干年的发展，1984 年国际标准化组织（ISO）开始对此提案进行讨论，并于 1986 年正式发布了为生成标准化文档而定义的标记语言标准（ISO 8879），称为新的语言 SGML，即标准通用标记语言。

SGML 功能非常强大，是可以定义标记语言的元语言，然而由于 SGML 过于复杂，不适合在 Web 上应用，因此，W3C 组织在 1996 年便开始设计一种可扩展的标记语言，以便能将 SGML 的丰富功能与 HTML 的易用性结合到 Web 应用中。1998 年 2 月，W3C 发布了 XML1.0 标准^①，其目的是为了在 Web 上能以现有的超文本标记语言（HTML）的使用方式提供、接收和处理通用的 SGML。XML 是 SGML 的一个简化子集，它以一种开放的、自我描述的方式定义了数据结构。在描述数据内容的同时能突出对结构的描述，从而体现出数据与数据之间的关系。

1.2 W3C 介绍

W3C 是万维网联盟（World Wide Web Consortium）英文的缩写，它成立于 1994 年 10 月，以开放论坛的方式来促进开发互通技术（包括规格、指南、软件和工具），开发网络的全部潜能。万维网联盟（W3C）从 1994 年成立以来，已发布了 90 多份 Web 技术规范，领导着 Web 技术向前发展。

W3C 认为自身不是官方组织，因此将它正式发布的规范称为**推荐（建议）标准**，意思是进一步标

^① 本书介绍的 XML 主要遵循 W3C 于 2004 年 2 月 4 日发布的 XML1.0 推荐标准的第三版，读者可在 <http://www.w3.org/TR/2004/REC-xml-20040204/> 了解到此标准的详细内容。

准化的建议，但是由于该组织自身的权威性往往成为事实上的标准。一项技术要成为 W3C 的推荐标准，需要经过以下 7 个步骤。

(1) W3C 收到提交

任何 W3C 的成员都可以提交一个建议。如果建议的内容在 W3C 的工作范围内，W3C 将决定是否要为此开展工作。

(2) W3C 发布注释

注释作为一个公开的文档，是对建议的描述。注释可以在发布后随时被修改或废弃。一个注释的发布，并不表示 W3C 对其认可了，注释仅供讨论使用。注释的发布，也不表示 W3C 已经开始任何与注释有关的工作。

(3) W3C 成立工作组

当一个提交被 W3C 认可之后，就成立由成员和一些对此感兴趣的团体参加的工作组。正常情况下，工作组将定义一个时间表，并发布一个提议标准的工作草案，描述当前工作进展。

(4) W3C 发布工作草案

W3C 通常会在其网站上 (<http://www.w3.org>) 发布工作草案，以及一个公众讨论的邀请。由于工作草案的内容可随时被修改、替换或废弃，所以不应把它作为工作的依据。

(5) W3C 发布候选推荐标准

当规范比较复杂时，可能需要成员和软件提供商花费更多的时间来试用或测试。有时候，这些规范以候选推荐标准的形式发布。它与工作草案一样都是进展中的工作文件，所以不应把它作为工作的依据。

(6) W3C 发布提议的推荐标准

提议的推荐标准的发布标志着工作组的工作到了最后阶段。提议的推荐标准仍然是一个进展中的工作文件，仍然可以被更新、替换或废弃。虽然提议的推荐标准没有被 W3C 正式认可，但是它在内容和时间上离最终的推荐标准已经非常接近了。

(7) W3C 发布推荐标准

推荐标准经 W3C 的成员审阅，并由 W3C 的主任加盖正式批准图章，而最终成为规范。W3C 的推荐标准是一个稳定的文档，可以作为工作中的参考资料。

W3C 组织于 2004 年 2 月 4 日，发布了 XML1.1 的推荐标准，这是最新的 XML 版本，不过目前大多数的应用还是基于 XML1.0 的推荐标准，因此本书也将遵照 XML1.0 规范来讲述。如果读者想要了解 XML1.1 规范的内容，可以参看网址：<http://www.w3.org/TR/2004/REC-xml11-20040204/>。

1.3 关于 XML 的几个问题

1. XML 是 HTML 的扩展吗

HTML 的全称是 Hypertext Markup Language (超文本标记语言)，而 XML 的全称是 (eXtensible Markup Language) 可扩展的标记语言，这很容易让人联想到 XML 是通过增加新标记来扩展 HTML 的一种标记语言。实际上 HTML 和 XML 在标记语言中处于不同的层次。

下面我们通过 HTML 文档和 XML 文档的对比（分别如例 1-1 和例 1-2 所示），来弄清楚 XML 的一些概念。

例 1-1 HelloWorld.html

```
<html>
  <head>
    <title>这是一个欢迎的例子</title>
  </head>
  <body>
    你好！欢迎你！
  </body>
</html>
```

例 1-2 HelloWorld.xml

```
<?xml version="1.0" encoding="gb2312"?>
<欢迎词>
  <标题>这是一个欢迎的例子</标题>
  <内容>你好！欢迎你！</内容>
</欢迎词>
```

从上面的两个例子，我们可以看出：

- ① 在编写 HTML 文档时，所有的标记都已经固定下来，我们不能去创造新的标记；而在编写 XML 文档时，我们可以任意地创建新的标记，包括中文的标记。所以说 XML 是可扩展的标记语言。
- ② 在编写 XML 文档时，没有一套标准的标记供我们选择使用，需要我们去创建标记，所以我们说 XML 是创建标记语言的元语言。



提示：XML 在设计之初，就考虑到了国际化的问题，同 HTML4.01 一样，XML 也是基于 ISO/IEC 10646 字符集标准中定义的通用字符集（Universal Character Set, UCS）。

2. SGML、HTML 和 XML 之间是什么关系

SGML 是一种在 Web 发明之前就早已存在的使用标记来描述文档资料的通用语言。它是一种定义标记语言的元语言。HTML 和 XML 都是从 SGML 发展而来的标记语言，因此，它们有一些共同点，如相似的语法和标记的使用。不过 HTML 是在 SGML 定义下的一个描述性的语言，只是 SGML 的一个应用，其 DTD（参见 1.8 节）作为标准被固定下来，而 XML 是 SGML 的一个简化版本，是 SGML 的一个子集，严格意义上来说，XML 仍然是 SGML。

HTML 不能用来定义新的应用，而 XML 可以，例如 RDF 和 CDF 都是使用 XML 定义的应用。事实上，XML 和 SGML 是兼容的，但又没有 SGML 那么复杂，它被设计用于有限带宽的网络，如 Internet。XML 规范的制定者之一 Tim Bray 说，XML 的设计出发点是取 SGML 的优点，去除复杂的部分，使其保持轻巧，可以在 Web 上工作。

HTML、SGML 和 XML 将继续用于其适合的地方，它们中的任何一个都不会使其他一个废弃。对于像新闻、网络日记、论坛留言等大部分短期的数据，HTML 仍是在 Web 上快速发布数据的最简单的方法。如果数据要长期使用，并且需要更多的一些结构，我们更推荐使用 XML。不同于 HTML 和 XML，SGML 可能永远不会在 Internet 上被广泛接受，因为它不是为某个网络协议而设计的，也从来没有为某个网络协议的需求而优化过。对于高端的、复杂结构的发布应用，SGML 将继续使用。

1.4 XML 与 HTML 的比较

通过 XML 与 HTML 的比较，我们能够更好地理解和掌握 XML 的优点。