

JSP 网站设计经典教程

吴其庆 编著

本书附盘可从本馆主页 <http://lib.szu.edu.cn/>
上由“馆藏检索”该书详细信息后下载，
也可到视听部复制

冶金工业出版社

2001 • 北京

内容简介

本书通过丰富的实例、详尽的 Java 和专业网络应用系统知识,介绍用 JSP 构建一个专业性网站的过程。主要内容包括 JSP 的基础知识、Java 的基础语法、JSP 内建对象和语法元素、JSP 的数据库操作,最后通过几个经典实例介绍了 JSP 在实际工作中的应用。

本书从实用性、易掌握性出发,重点突出、操作简练、内容丰富,同时附有大量的操作实例和练习,读者可以一边学习,一边在电脑上操作,做到理论与实践相结合,因此本书既是 JSP 初学者的培训教材,又是网页设计专业人士理想的参考书。

图书在版编目(CIP)数据

JSP 网站设计经典教程 / 吴其庆编著. - 北京:
冶金工业出版社, 2001.4
ISBN 7-5024-2759-7

I. J... II. 吴... III. 网站-设计-教材
IV. TP393.092

中国版本图书馆 CIP 数据核字(2001)第 16444 号

JSP 网站设计经典教程

吴其庆 编著

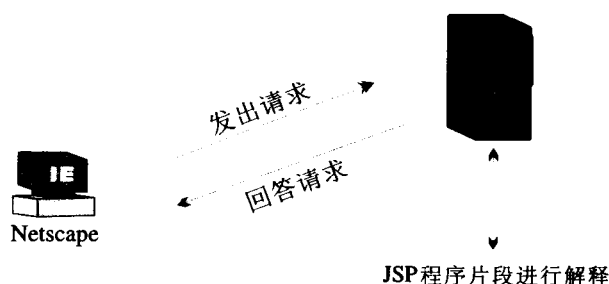
出版	冶金工业出版社
社址	北京沙滩嵩祝院北巷 39 号, 邮编 100009
发行	冶金工业出版社发行; 各地新华书店经销
印刷	广东出版技校彩印厂印刷
开本	787×1092 毫米 1/16
印张	28
字数	643 千字
版本	2001 年 4 月第 1 版 2001 年 4 月第 1 次印刷
印数	1-1900 册

书号 ISBN 7-5024-2759-7 / TP·232
定价 45.00 元(含磁盘)

前言

JSP (Java Server Pages™) 是在 Sun Microsystems 公司倡导下建立的一种动态网页技术标准 (参见 <http://www.javasoft.com/products/jsp>)。它在运行速度、安全性、可移植性均比传统的 Web 编程语言有很大的改进。“一次开发, 多个平台应用”更是 JSP 动人之处, 因此深受 Web 开发人员的青睐。

JSP 与微软的 ASP (Active Server Pages) 相似, 但它使用类似 HTML 的标签以及 Java 程式码片段, 而不是 VBScript。在传统的 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, 然后将结果以 HTML 格式返回给客户。具体的执行过程如下图所示。



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。所有程序操作都在服务器端执行, 从网络传送给客户端的仅是得到的结果, 对客户浏览器的要求最低, 可以实现 No Plugin、No ActiveX、No Java Applet、No Frame。

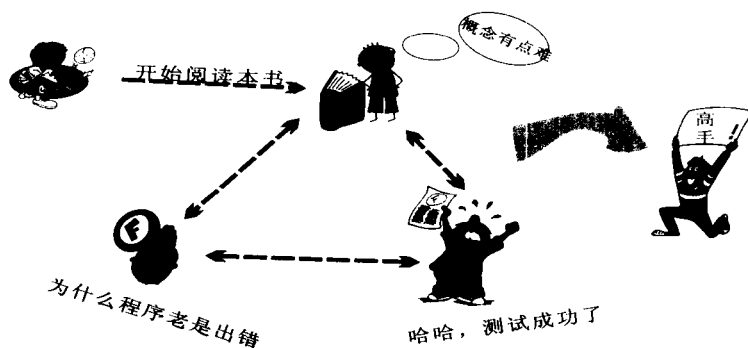
当你所使用的网站服务器没有提供本地 ASP 支援, 如 Apache 或 Netscape 服务器, 那么你可以考虑使用 JSP。虽然你可以取得这些服务器的 ASP 附加配备模块, 但价格却相当昂贵, 而目前 Sun 并没有对你所需要的 JSP 元件收取费用 (虽然 Sun 将来有可能会收费)。对于 Solaris 与 Linux 以及 Windows, 这些控件也都很容易取得。

JSP 支持众多 Web 服务器, 包括 Apache Web Server 和 IIS 4.0 在内的 85% 的服务器产品。正因为 JSP 有极好的兼容性, 当 Sun 正式发布 JSP 后, 其用户就“一见钟情”, 用户总量也在不断地提高。

JSP 与 Microsoft 的 ASP 技术非常相似。两者最明显的区别是 ASP 的编程语言是 VBScript 之类的脚本语言, 而 JSP 使用的是 Java。它们的本质区别是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, 这样便充分发挥了对象的共享性。事实证明, Java Servlet 是一种开发 Web 的理想构架。JSP 以 Servlet 技术为基础, 又在许多方面作了改进。利用跨平台运行的 JavaBean 组件, JSP 为分离处理逻辑与显示样式提供了卓越的解决方案。JSP 必将成为 ASP 技术的有力竞争者。

本书从实际的编程出发, 介绍了如何使用 JSP 这个美妙的工具。本书基本概念清晰,

并结合具体实例进行操作,让读者以尽快的速度从入门到精通,如下图表示。



本书按照由浅入深的原则,循序渐进地介绍了 JSP 的各项功能和使用方法:

首先介绍了 JSP 的基本概念,通过基本概念的讲解,使读者对 JSP 有一个总体认识。接着通过实例剖析,详细介绍了 JSP 的使用方法和使用技巧。最后介绍了如何使用 JSP 建立网站。

为便于读者学习,本书附送一张软盘,本软盘包括本书所有源程序文件,共有 337 个文件,超过 30000 条的源代码,用户不需要输入源程序便可以方便地从代码中快速学习 JSP,同时用户可以免费获得 10 个综合系统的源代码,如在线商店、在线考试、在线调查区等,这些系统对购买本书用的户来说,绝对好用与超值。软盘的使用说明附在本书后。

本书对没有 JSP 编程基础,但懂得 C、Java 者更容易入门。因此适合于 JSP 初学者,对网页设计专业人士也是一本很好的参考资料。为进一步提高读者对 JSP 的应用技能,书中给出了一定量的练习,并附有参考答案。

人无完人,本书也不敢狂称十全十美,但求可以给如我一样爱好 JSP 的人一些帮助,编者即感到十分高兴了。我将这一本书送给我所爱的父母亲及我最好的知己 Ally He。

在阅读的过程中,如果遇到什么困难或本书有错漏的地方,请 E-mail: qibrother1116@263.net 进行探讨。

编 者

2001 年 3 月

第一章 JSP 概述

本章主要介绍了 JSP 基础知识, 包括 JSP 与 JavaBean 的关系、JSP 的优缺点、JSP 和 ASP、PHP 的比较及其应用前景等。

1.1 JSP 简介

每当人们接触一项新的技术时, 总希望在最短的时间去了解它, 使用它, 精通它。那么本书正是为这样的需求而设计。当接触一门新的技术, 初学者总会有一大串陌生的问题、总爱在自己已存在的记忆中寻找与之相近的内容, 如精通 ASP 与 Visual Basic 的程序员第一个念头便是“为什么要学习 JSP? JSP 可以实现的一切, ASP 也可以做到吗?”, 接着又一个念头“到底要花多少时间才可掌握 JSP?”; 用 Delphi 进行多层程序设计的程序员第一个念头“JSP 是否可以分布式管理, 效率怎么样? 要知道 Delphi 是以效率著称的。”; PHP 及 Perl 的使用者的第一念头很可能便是“JSP 是否可以在 Unix、Linux 上执行。”下面就针对上述问题, 对 JSP 进行全面系统地介绍。

1.1.1 JSP 的实质

JSP 的实质便是 Servlet, 更准确一点便是, 在 Web 服务器 (或是 Servlet 服务器、Application Server, 服务器是户请求的服务中心) 中解释为 Servlet, 这时候 JSP 已变成 Java 程序员熟悉的 Servlet 来执行, 接着它被编译成 Java 的执行代码 (bytecode), 使用标准的 API 执行 Java Servlet, API 便是 JSP 的核心所在, 这个解释过程也导致了 JSP 在第一次编译时有点慢。

Servlet 与 JSP 的代码对比如图 1-1 所示。

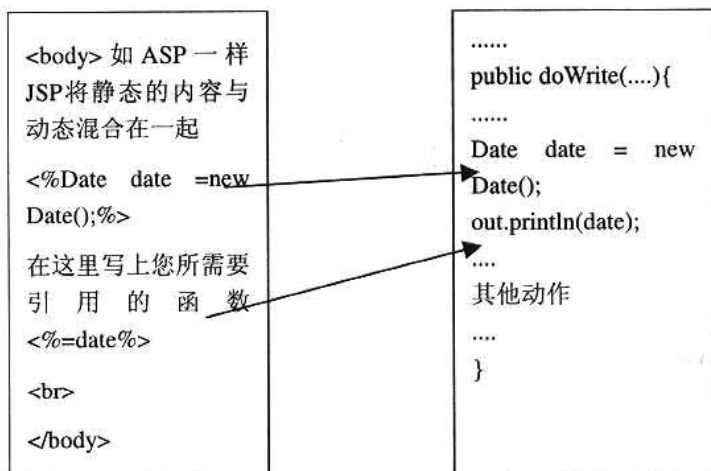


图 1-1

既然 JSP 就是 Servlet, 那么为什么还要有 JSP 的出现? JSP 主要是针对 Java 程序处理

不熟悉的人而设计的。

1.1.2 JSP 的优势

与 Servlet 相比, JSP 有如下两个优势:

1. 方便与 HTML 混合

对 HTML 来说, JSP 容易对 Java 插入对象, 使 HTML 更容易增加动态部分, 而 Servlet 却要对 HTML 的输出做特殊的处理, 增加了代码的编写测试时间 (如 “\” 这类跳脱字元), 遇上复杂的 HTML 网页编写, 用 Servlet 简直就是一场恶梦。

2. 在开发、测试方面比 Servlet 方便

修改 JSP 后, 马上就可以看到执行的结果, 不需编译; 而 Servlet 却要编译, 放入执行代码等复杂过程, 将 HTML 的代码与 Java 的程序代码混合在一起增加了除错的难度。

本书的重点放在 JSP 的对象及语法分析上, 略讲 Servlet 的内容, 如果读者要进行更高级的程式处理, 请参考有关 Java 方面的书籍。看到这里, 心急的读者难免一阵心跳, 原来学习 JSP 还要掌握 Java 方面的知识。正因为 JSP 与 Java 紧紧连在一起, 才使 Java 显示出无限的活力。作为一名程序员总想学一门一步到位的程序处理, 但每当一门新的技术公布后, 都要痛苦地收集各方面的资料。

通过上面的说明, 我们已经明白了 JSP 与 Servlet 关系与各自扮演的角色, 当一个用户在其客户端发出请求时, JSP 的执行过程如图 1-2 所示。

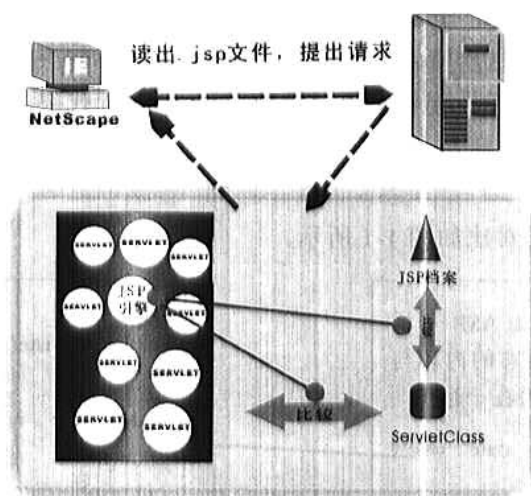


图 1-2

1.2 JSP 与 JavaBeans 的关系

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 (内容是根据请求而变化的, 例如请求登录者的信息)。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，能够方便地编辑和使用 JSP 页面，而不需要理会 JavaBeans 到底进行什么操作。

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。

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，或者使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。

如果要进行电子商务网站的设计或更复杂的应用程序（网上 MIS——Management Information System）设计，那么 JSP 的复杂架构必须由 JavaBean 来达成，这种架构在其他语言想要整合是十分困难或是缺乏经济效益的，因此不会有超出 Java 形式存在的 JSP，最多是以非 Servlet 的形式运作，但仍然是架构在 Java 上面。

如图 1-3 所示清晰地表示上述的关系。

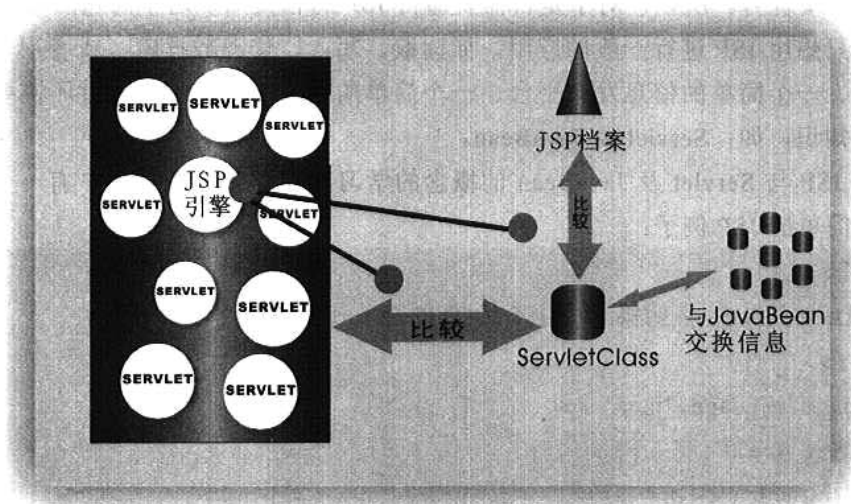


图 1-3

为了使读者更好地了解 JavaBean 的概念，下面列举一个用 Java 写 JavaBean 的例子。

首先，一个 JavaBean 必须实现可序列化的接口，支持这个接口的对象可以存储和重建它们的状态。

如下所示：

```
Public class Simplebean extends Canvas implements Serializable
```

然后，我们可以为这个 JavaBean 定义属性和事件，这样这个 JavaBean 就可以具有一定的功能了，如下：

```
private Color beanColor = Color.blue;
```

然后再定义如下两个方法，这两个方法需要定义成 public 类型：

```
public Color getColor(){
    return beanColor;
}
public void setColor(Color newColor){
    beanColor=newColor;
    repaint();
}
```

同时，为了使 beanColor 属性能够与 JavaBean 的表现联系起来，需要重载 Canvas 的 paint()：

```
public void paint(Graphics g){
    g.setColor(beanColor);
    g.fillRect(0,0,width,height);
}
```

从 JSP 与 Servlet 及 JavaBean 的关系可以看出，JSP 的优势在于 Java 所存在的强力 API 与企业支持而非 JSP 语法本身，JSP 语法本身与其他 Server-side Script 比较，并没有突出之处，也并非最优秀。所以，对于是否需要懂得 JSP 的相关知识这个问题回答如下：

如果您只想用 JSP 进行一般的应用，如读取、写入、修改数据库，或者用 JSP 开发一个个人主页、一个简单的信息发布平台、一个简单的购物平台，那么您并不需要深入学习 Java 方面的知识，如：Servlet 及 JavaBean。

通过对 JSP 与 Servlet 及 JavaBean 的概念的学习，为了使读者对 JSP 有一个清晰的概念，举一个简单的 JSP 例子：

```
< HTML>
< HEAD>< TITLE>我的第一个 JSP </TITLE>< /HEAD>
< BODY>
< %@ page language="java" %>
< %! String str="0"; %>
< % for (int i=1; i < 10; i++) {
    str = str + i;
} %>
JSP 输出之前。
< P>
< %= str %>
< P>
JSP 输出之后。
< /BODY>
< /HTML>
```

读者可将这个代码直接将代码放在一个 .jsp 文件中进行测试，如图 1-4 所示为 Homesite

编辑软件。



图 1-4

其结果经 TOMCAT 的解释如图 1-5 所示。

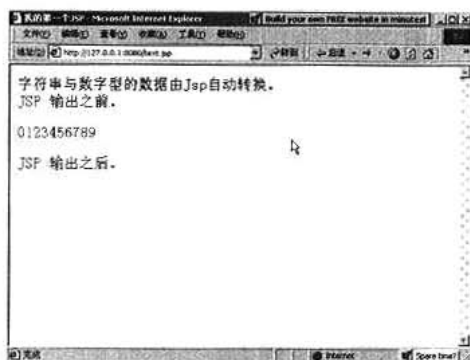


图 1-5

这个 JSP 页面可以分成几个部分来分析。

1) JSP 指令。它描述的是页面的基本信息，如所使用的语言、是否维持会话状态、是否使用缓冲等。JSP 指令由 `< %@` 开始，`%>` 结束。在本例中，指令“`< %@ page language="java" %>`”只简单地定义了 Java 语言（当前，在 JSP 规范中 Java 是惟一被支持的语言）。

2) JSP 声明。JSP 声明处可以看成是定义类这一层次的变量和方法的地方。JSP 声明由 `< %!` 开始，`%>` 结束。如本例中的“`< %! String str="0"; %>`”定义了一个字符串变量。在每一项声明的后面都必须有一个分号，就像在普通 Java 类中声明成员变量一样。

位于 `< %` 和 `%>` 之间的代码块是描述 JSP 页面处理逻辑的 Java 代码，如本例中的 for 循环。

3) 位于 `< %=` 和 `%>` 之间的代码称为 JSP 表达式，如本例中的“`< %= str %>`”。JSP 表达式提供了一种将 JSP 生成的数值嵌入 HTML 页面的简单方法。

1.3 JSP 的技术评价

1) JSP 编译成 Servlet，用户在第一次访问这个页面的时候所花的时间较长，后面的用

户直接从内存中读出数据。解决办法：开发者成为第一个用户，开发者为后面的用户铺路。

2) Servlet 仍然是解释执行的。Java 虚拟机本身就是解释执行的。

3) JSP 虽是 Servlet 的延伸，但与 Servlet 还有很大不同。尤其是 JSP 1.0 以后。GNU JSP 采用的是 0.92 的标准，对 Bean 的支持比较差。

4) 在美国，EJB+Servlet+JSP 几乎成为电子商务的开发标准，它将成 ASP 的强有力竞争对手。PHP 由于其模式和一些天生缺陷，毫无希望。在美国，很少有商业站点用 PHP 的。

1.4 JSP 的强势

1) 一处编写，随处运行。这是一个程序员的梦想，也是从前的程序员的恶梦，为了在不同的平台间运行，使许多程序员一行行的重写代码。在这一点上 Java 已经给出了最完美的答案，它要比 PHP 更出色，除了系统之外，您的代码不用做一点更改的。

2) 系统的多台平支持。已经在差不多所有平台上见到了一致的 Java Servlet/JSP 的 Web Server/Application Server 系统，它可以让您在任意环境进行开发、系统部署及扩展。相比 ASP/PHP 的局限性是显而易见的。

3) 强大的可伸缩性。从只有一个小的 jar 文件就可以运行 Servlet/JSP，到由多台服务器进行集群和负载均衡，以及多台 Application 进行事务处理、消息处理……一台服务器到无数服务器，Java 都显示了一个巨大的生命力。

4) 多样化和功能强大的开发工具支持。这一点与 ASP 很像，Java 已经有了许多非常优秀的开发工具，如 JBuilder3、Homesite、Visual Java，而且许多可以免费得到，并且可以顺利的运行于多种平台之下。任何工具的强大功能都建立在您是否熟练掌握它们，只有通过日积月累的学习，才可以显示它们的威力。

5) 现在的企业应用已从 2-tier 向 3-tier 结构转移，开发者有许多不同的方法分发应用与数据到用户。JSP API 是 J2EE 的一部分，给 Web 和 Java 开发者创建各种应用提供了一个简单但强有力的机制，JSP 给开发者提供了两个重要的能力：

① 它提供了访问远程数据的机制如 EJB, RMI, JDBC。

② 它让开发者从一堆代码中分离程序逻辑而达到最大程度的代码重用和灵活性。逻辑与表现的分离是它超越其他 Web 应用结构的一个主要优势(随着 COM/DCOM 模形的成熟，Delphi 在分布式程序管理方面也有杰出的表现)。

6) Java Servlet 是一种开发 Web 应用的理想构架。JSP 以 Servlet 技术为基础，又在许多方面作了改进。利用跨平台运行的 JavaBean 组件，JSP 为分离处理逻辑与显示样式提供了卓越的解决方案。JSP 必将成为 ASP 技术的有力竞争者。

1.5 JSP 的劣势

1) 与 ASP 一样，Java 的一些优势正是它致命的问题所在。由于为了跨平台的功能，为了极度的伸缩能力，所以极大的增加了产品的复杂性。它在扩展时需要分成多少块，那么 Java 系统中就有多少种产品。所以能够看到 JRE、JDK、J2EE、EJB、JSWDK……等一系列名词，实际上它们只要有有效的搭配在一起，就可以产生强大的功能，密不可分。

2) Java 的运行速度是用户将 class 常驻内存来完成的, 所以它有时所使用的内存和用户数量相比, 是“最低性能价格比”。另一方面来说, 它还需要硬盘空间来存储一系列的 .java 文件和 .class 文件以及对应的版本文件。

3) JSP 只有对 Java 相当熟悉的人才能用得很好。

4) JSP 程序调试困难。JSP 页面首先被转化为一个 .java 文件 (Servlet), 然后再被编译。这样, 出错信息实际上指向的是经过转化的那个 .java 文件而不是 JSP 本身。例如在 JSP 文件第一行的错误可能被显示为第 20 行。如果试图通过反跟踪来解决这个问题有时会使结果显得很滑稽 (但是, 在 Resin 的引擎中, 错误已经可以指向 JSP 文件的出错行)。也可以使用一些集成设计工具 IDE 如 VAI 或 JDeveloper, 但这有悖于 JSP 简单易用这一基本特性。

5) 数据库连接并不那么容易。大多数 Servlet 引擎并不支持本地的缓冲池连接。因此, 不得不写很多代码来完成这项工作。

6) 很难选择合适的 Servlet 引擎。没有一个独立的组织来对各种引擎提供独立的 benchmark 测试。提供商当然也对他们自己的产品提供了 benchmark 测试, 但结论通常以数据形式给出, 且均回避了稳定性。

7) 在 JSP 编程时问题很多, Jason Hunter 《Java Servlet programming》一书的作者写过一篇有趣的文章比较了 JSP 和其他替代工具。可参看 <http://www.servlets.com/soapbox/problems-JSP.html>。

1.6 JSP 与 ASP 的比较

从微软推出了 ASP (Active Server Page) 后, 它以其强大的功能, 简单易学的特点而受到广大 Web 开发人员的喜欢。但是它却有微软产品的通病, 只能在 Windows 平台下使用, 虽然它可以通过增加控件而在 Linux 下使用, 但是其功能最强大的 DCOM 控件却不能使用。而 SUN 公司在 Java 的基础上开发出的 JSP (Java Server Pages) 实现了动态页面与静态页面的分离, 脱离了硬件平台的束缚, 以及编译后运行等方式, 大大提高了其执行效率而逐渐成为因特网上的主流开发工具。

1.6.1 比较项目

现把其异同对比如下:

1. 结构

JSP 和 ASP 在结构上类似, 都是以 " " 作为标识符, 不同的是在标识符之间的代码。ASP 为 JavaScript 或 VBScript 脚本, 而 JSP 为 Java 代码。JSP 将网页的表现形式和服务器的代码逻辑分开。

作为服务器进程的 JSP 页面, 首先被转换成 Servlet (一种服务器端运行的 Java 程序), Servlet 支持 HTTP 协议的请求和响应。当 JSP 被转换成纯 Java 代码。多个用户同时请求一个 JSP 页面时, 应用实例化线程来响应请求。这些线程由 Web 服务器进程来管理, 和 ASP 的线程管理器功能类似。同 CGI 为每个请求创建一个进程的模式比较, 这种方式效率高得多。

2. 性能和平台无关性

与 C++、Visual Basic 等语言比较, Java 被看作是个效率不太高的语言, 但它以牺牲效率换来了平台无关性, 使 Java 可以在大多数操作系统上运行, 而不需要重新编译——许多人愿意为此而付出代价。Java 的编译代码是一种字节代码, 在运行时由操作系统上的一个 Java Virtual Machine (JVM) 虚拟机解释。字节代码可以在所有平台上迁移, 而不需要任何改动。交叉平台编程倾向于采用所有操作系统的“最小公分母”, 尤其是使用图形化界面时 (Servlets 不需要任何图形界面)。Servlets 比传统的 Java 程序 (Applets, Javaapp) 要快, 因为它们在服务器端运行, 不需要加载沉重的 GUI (HTML 的 GUI 是非常少的)。另外, Servlets 的字节代码只有在客户请求时才执行, 所以尽管当首次调用 Servlets 时会有几秒钟的加载时间, 但后续的请求相应非常迅速, 因为服务器已经缓存了运行的 Servlets。当前的 JSP 服务器都带有 Java 即时编译器 (JIT), 因此, JSP 的执行比每次都要解释执行的 ASP 代码要快, 尤其是在代码中存在循环操作时, JSP 的速度要快 1 到 2 个数量级。

3. Session 管理

为了跟踪用户的操作状态, ASP 应用 Session 对象。JSP 使用一个叫 HttpSession 的对象实现同样的功能。Session 的信息保存在服务器端, Session 的 id 保存在客户机的 cookie 中。如果客户机禁止 cookie, Session id 就必须带在 URL 后面。Session 一般在服务器上设置了一个 30 分钟的过期时间, 当客户停止活动后自动失效。Session 中保存和检索的信息不能是基本数据类型 (primitivedatatypes) 如 (int, double) 等, 而必须是 Java 相应的 object (对象) 如 (Integer, Double)。

```
HttpSession session=request.getSession(); //得到一个 session 对象或创建一个
session.getId()//得到 sessionID 数
Integer id=(Integer)session.getValue("id") //检索出 session 的值并转化为整型
session.putValue("ItemValue",itemName); //ItemValue 必须不是 must 简单类型
```

4. Application 管理

有时服务器需要管理面向整个应用的参数, 使每个客户都能获得同样的参数值。和 Session 一样, ASP 使用 Application 对象, 而 JSP 使用 Servlet Context 对象, 或者 Application 内建对象, 操作的方法和 Session 一样。

```
getServletContext().setAttribute("Item",ItemValue); //设置一个应用变量
Integer i=(Integer)getServletContext().getAttribute("ItemName"); //得到//item
Application.setAttribute("Item",ItemValue) //建立一个内建对象
```

5. ServerSideIncludes

在服务器端的引用上 ASP 和 JSP 有着相同之处。ASP 和 JSP 都可以在支持此功能的服务器 (IIS, APACHE) 上实现服务器端包含虚文件。但 JSP 是将动态页面的结果包含进来, 而不是包含文件的代码本身。当包含的文件在另一个服务器上时, 不包含任何代码和对象是一个非常有效的功能。

```
ASP:<%# include file="subfile.JSP"-->
JSP:<%@include file="subfile.JSP"-->
```

6. Java 组件: JavaBeans

JavaBeans 是一些完成预定义功能的封装对象数据。JavaBeans 和 JSP 结合与 COM 和 ASP 相比如下:

COM 对象常用来封装商业逻辑和为 ASP 页面完成高强度计算。重用的组件使页面简单快速, 因为组件由编译语言 (C++、Visual Basic、Delphi) 构成, 而不是解释型的 Scripting 语言 (VBScript, JScript)。

JavaBeans 只能用 Java 语言开发, COM 可以由符合标准对象模型的任何语言开发 (包括 Visual J++)。另一方面, JavaBeans 更容易开发, 因为一旦掌握了 Java, 了解 JavaBeans 的结构非常容易。由于 COM 的复杂性, 需要大量的学习和培训才能掌握, 即使您具有 C++ 和 Visual Basic 的基础也需要额外的努力。COM 组件需要在服务器上注册, 如果您修改了一个现有的组件, 服务器需重新启动才能使用它。JavaBeans 不需要重新注册, 如开发者不需得到对服务器的完全访问权限, 这是其一个巨大的优点。可创建满足商业逻辑的完整的 JavaBeans 库, 让非编程人员使用库来开发动态网站。Beans 可在服务器端管理数据库连接。

JavaBeans 符合结构化对象模型: 每个 Bean 由一系列函数设置其属性。要设置 Bean 的内置属性, 必须使用带有属性名的 setProperty 标签。如果一个属性可以设置, Bean 需要有一个 setXxxx 方法, Xxxx 用实际的属性名来代替。

7. JSP 和数据库

ASP 使用 ODBC 通过 ADO 连接数据库, 而 Java 通过一个叫 JDBC 的技术连接数据库。目标数据库需要一个 JDBC 驱动程序, 一个数据库与 Java 的接口, 使 Java 可以用标准的方式访问数据库。JDBC 不使用服务器端的数据源。只要有 JDBC 驱动程序, Java 就可以访问数据库了。如果一个特定的数据库没有 JDBC 驱动程序, 而只有 ODBC 驱动程序, Java 提供一个 JDBC-ODBC 桥来将 JDBC 调用转化为 ODBC 调用。所有的 Java 编译器都带有一个免费的 JDBC-ODBC 桥。理论上, 桥可以访问任何常见的数据库产品。

1.6.2 结论

JSP 模型是在 ASP 之后定义的, 它借用了 ASP 的许多优点, 如 Session, Application 等对象。

同时 JSP 使用灵活而强大的 Java 语言, 而不是低效的 Scripting 语言。ASP 开发者只能使用基于 Windows 平台的技术, 而 Java 和 JSP 是跨平台的。

1.7 JSP、ASP、PHP 的测试报告

在 ASP、PHP 盛行的今天, JSP 在执行效率方面是否有优势?

下面是第三方的测试数据:

首先测试的是 JSP, 测试采用 20000×20000 次循环运算, 测试结果令人不可思议, 该程序只用了不到 4 秒的时间就运行结束了。

程序代码如下:

```
<JSP:useBean id='clock' scope='page' class='dates.JSPCalendar' type="dates.
```

```
JSPCalendar"/>
<JSP:getProperty name="clock" property="time" />
<%int i=0;
int j=0;
int k=0;
for(i=0;i<20000;i++){
    for(j=0;j<20000;j++){
    }
}%>
<JSP:useBean id='clock1' scope='page' class='dates.JSPCalendar' type=" dates.
JSPCalendar" />
<JSP:getProperty name="clock1" property="time" />
```

下一步测试的是 PHP，最初也准备测试 20000×20000 次，但是不管把程序最大运行时间调得再大，也无法运行结束，因此改为 2000×2000 次，运行时间竟然达到 84 秒。

程序代码如下：

```
<?TTime=date("Y-n-d"). " " . date("H:i:s");
echo $TTime;
echo "";
for($i=0;$i<1000;$i++){
    for($j=0;$j<1000;$j++){
    }
}
$TTime=date("Y-n-d"). " " . date("H:i:s");
echo $TTime;?>
```

最后进行的是 ASP 的测试，测试的循环也是 2000×2000 次，运行时间是 63 秒。

程序代码如下：

```
<%
response.write now
response.write "<br>"
for i=0 to 2000
    for j=0 to 2000
    next
next
response.write now
%>
```

另外，对数据库连接的测试如下：对 Oracle 8 进行 1000 次 Insert, Update, Select, 和 Delete。JSP 需要 13 秒，PHP 需要 69 秒，ASP 需要 73 秒。

1.8 JSP 的应用范围

电子商务网站设计, EJB+Servlet+JSP 成为了电子商务网站的标准范式。

用户网站的制作与整合。

企业网的信息管理。

企业办公自动化的应用。

应用于需要跨平台的网站。

1.9 JSP 的技术前景

JSP 技术被设计为一个开放的、可扩展的建立动态 Web 页面的标准。开发人员可以使用 JSP 页面来创建可移植的 Web 应用, 在不同的 Web 和应用服务器上为不同的场合所运行, 而不论采用什么适合本身场合和需要的创建工具。

通过与业界领域的合作, Sun 保证 JSP 规范是开放的和可移植的。可以使用任何客户机和服务器平台, 在任何地方都可编写和部署它们。将来, 工具供应商和其他厂商将通过为专门的功能提供客户化的标识库而扩展平台的功能。

JSP 规范的 1.0 版本是通向动态 Web 页面生成的一个开放的工业标准方法的第一步。1.0 版本通过一个核心标识集、隐含对象以及开始创建动态 Web 页面所需的基本功能构成了该方法的基础。已经有几个 Web 服务器, 应用服务器和开发工具供应商正在为他们的产品添加 JSP 1.0 的支持, 这样在业界已经拥有了最初的、立即的支持。

最新的一个版本将通过更多的 XML 支持、客户化标识、以及与 J2EE (Java2 Platform Enterprising Edition) 的集成而扩展这个版本。并且供应商们可能会有选择地扩展和扩充在 JSP 规范中基本的、必需的功能。JSP 引擎能够强有力地支持多种脚本语言和对象模型。在业界扩充和使用 JSP 技术能力的同时, 也向 Sun 承诺将保证 JSP 技术保持平台和服务器间固有的可移植性。

1.10 JSP 的专有名词解释

1) Application Programming Interface (API): 应用函数, 也称程序应用界面, 如 Java 核心程序库。

2) Application Scope: 应用程序的生命周期。

3) Client: 客户端, 如用户使用的 IE 及 Netscape 等, 如果不明白这个概念, 就会像很多初学者所提出的问题, 为什么要双击.asp 及.php, 又或者是.JSP 文件运行不了呢?

4) Content Type: 内文格式。

5) FTP (File Transfer Protocol): 文件传输格式, 如在网站下载文件时多采用这个格式。

6) HTML (Hyper Text Makeup Language): 定义网页内容的语言, 又称超文本。

7) HTTP (Hype Text Transfer Protocol): 超文本传输协议, 使文本、图片、影像在网络上传递。

8) IP (Internet Protocol): 互联网传送协议。

9) JAR (.jar): Java Archive 是 Java 平台的内建压缩格式, 能有效地将资源整合。

10) JavaBeans: 跨平台并且可重用的软件。

11) Java Database Connectivity (JDBC): 提供 Java 平台与数据连结能力, JDBC 提供了一组 API 与 SQL 语法存取方法。

12) JavaScript: 一种动态网页的通用脚本语言, 如大量在 DHTML 应用。

13) JSP element: 包括指令语法、Scripting 与动作语法。

14) Multi-purpose Internet Mail Extensions: 一种用来传送多媒体信息格式 E-mail 与信息的格式。

15) Server: 提供服务的地方。

16) Session Scope: 一个访问客户的生命周期, 一般来说, 一个 Session 便是从您登录一个网站到您离开一个网站的时间。

17) TCP (Transmission Control Protocol): 传送控制协议, 实现互联网环境中的可通信而使用的主机到主机协议。

18) URI (Uniform Resource Identifier): 用来定位某种网络协定上的资源文法格式, URL 是 URI 的一个子集。

19) URL (Uniform Resource Location): 用来显示网络某个文件的位置, 如: [Http://www.sina.com.cn](http://www.sina.com.cn)。

20) WWW (World Wide Web): 万维网的总称。

总结: 本章简明地介绍了 JSP 的技术基础, 如技术的优势、劣势、执行效率、技术展望等。

综合练习一

基础题

一. 填空题

1. JSP 的实质是_____。
2. 电子商务的开发标准是_____。
3. JSP 的英文全称是_____。

二. 判断题

1. 学习 JSP 并不需要学习 JavaBean。
2. JSP 的执行肯定要经过 Servlet 才能执行。
3. JSP 就是 ASP。
4. JSP 不带 Session 的管理。
5. JSP 不可以制作聊天室。
6. JSP 不带 Application 处理。
7. 在 JSP 开头并不需要 `<%@ page language="java" %>` 这句话。

三. 简答题

1. 与 Servlet 相比, JSP 具有哪些明显的优势?
2. JSP 有哪些特点?
3. 在编程方面, JSP 有哪些不足?
4. 请解释下面英文的意思。

Application Programming Interface(API)

application scope

Client

content type

FTP(File Transfer Protocol)

HTML(Hyper Text Makeup Language)

HTTP(Hype Text Transfer Protocol)

IP(Internet Protocol)

JAR(.jar)

JavaBeans

Java Database Connectivity (JDBC)

JavaScript

JSP element

Multi-purpose Internet Mail Extensions

Server

session scope

TCP(Transmission Control Protocol)

URI(Uniform Resource Identifier)

URL(Uniform Resource Location)

WWW(World Wide Web)

5. 画出 JavaBean 与 JSP 的关系图。

上机实习

试编写一段 JSP 小程序, 熟悉 JSP 操作方式。