

网络建设
与开发丛书

JSP 核心技术

精英科技 编著



中国电力出版社

www.infopower.com.cn

网络建设与开发丛书

JSP

核心技术

精英科技 编著

中国电力出版社

内 容 提 要

本书是一本关于系统学习 JSP 的参考书籍, 全书在力求详尽介绍 JSP 基础知识的同时, 侧重于使用技巧和高级应用。本书共分十一章, 前两章介绍 JSP 基础知识, 第三章到第八章介绍 JSP 的核心技术, 第九章到第十一章介绍了三个实例, 具有极强的代表性和可操作性。

本书既适用于 JSP 初学者, 又适合于高级的 JSP 编程人员。

图书在版编目 (CIP) 数据

JSP 核心技术/精英科技编著. -北京: 中国电力出版社, 2001
ISBN 7-5083-0605-8

I. J... II. 精... III. 计算机网络-程序设计 IV. TP393.092

中国版本图书馆 CIP 数据核字 (2001) 第 23506 号

中国电力出版社出版、发行

(北京三里河路 6 号 100044 <http://www.infopower.com.cn>)

三河市实验小学印刷厂印刷

各地新华书店经售

*

2001 年 6 月第一版 2001 年 6 月北京第一次印刷

787 毫米×1092 毫米 16 开本 23 印张 514 千字

定价 35.00 元

版 权 所 有 翻 印 必 究

(本书如有印装质量问题, 我社发行部负责退换)

前言

在 Sun 正式发布 JSP (Java Server Pages) 之后, 这种新的 Web 开发应用技术很快引起了人们的关注。JSP 为创建高度动态的 Web 应用提供了一个独特的开发环境。据 Sun 公司说, JSP 能够适应市场上包括 Apache Web Server、IIS4.0 在内的 85% 的服务器产品。即使您习惯了使用微软公司的 ASP, 关注 JSP 的发展仍然对您很有帮助。

本书是一本系统学习 JSP 的参考书籍。本书共分 11 章。各章内容分别如下:

第一章 初识 JSP。介绍 JSP 背景、特点及其发展前景。

第二章 安装 JSP 的使用环境。介绍在 Windows 和 Unix 环境下安装 JSP 开发环境的方法, 并提供了第一个 JSP 例子。

第三章 简明 Java 语法。介绍 Java 语法, 这是编写 JSP 程序的基础。

第四章 JSP 的简明语法。介绍 JSP 的语法。

第五章 灵活的 JSP 动作标识。介绍 JSP 的 5 个核心标识。

第六章 JSP 中的字符串处理。介绍 JSP 中处理字符串的方法。

第七章 JSP 与后台数据库管理。介绍 JSP 与后台数据库的关系。

第八章 Servlet 编程。介绍 Servlet 编程的技巧。

第九章 制作日历。介绍日历这个实用小例子的编写方法。

第十章 留言板。介绍留言板的编写方法。

第十一章 网上书店。介绍电子商务实例——网上书店的编写方法。

从上述内容我们可以看出本书包括了 JSP 的所有核心内容, 同时还有大量的实例。因此本书适用范围很广, 它既适用于初学者, 也适用于高级的 JSP 编程人员。作为一本介绍 JSP 核心技术的书籍, 本书力求在详尽介绍 JSP 基础知识的同时, 侧重于使用技巧和高级应用。同时它也是一本参考书, 本书附录包含了 SQL、html 和 JavaBean 的语法列表。

由于时间仓促, 加之水平有限, 书中缺点错误在所难免, 恳请广大读者批评指正!

作者

2001 年 3 月

目 录

前 言

第一章 初识 JSP	1
1.1 开发基于 Web 的应用程序: 背景	1
1.2 JSP 的特点	2
1.3 JSP 页面的应用模型	3
1.4 简单快速地编程	4
1.5 JSP 页面与 XML 技术的集成	4
1.6 JSP 技术的未来	5
第二章 安装 JSP 的使用环境	6
2.1 在 Windows 下的安装与设置	6
2.2 在 Linux 下的安装与设置	12
2.3 一个简单的 JSP 程序	18
第三章 简明 Java 语法	20
3.1 初识 Java	20
3.2 Java 代码的基本格式	20
3.3 Java 关键字	22
3.4 Java 标识符	24
3.5 变量和常量	24
3.6 Java 基本数据类型	26
3.7 程序流程	31
3.8 运算符	44
3.9 面向对象基础	53
3.10 数组类型	68
3.11 异常处理	72
第四章 JSP 的简明语法	84
4.1 JSP 指令	84

4.2	include 指令语法	93
4.3	JSP 声明	95
4.4	JSP 表达式	96
4.5	JSP 脚本元件	97
4.6	JSP 注释	98
4.7	一些特殊字符串的写法	100
4.8	JSP 隐含对象	102
第五章	灵活的 JSP 动作标识	112
5.1	useBean 的属性	112
5.2	setProperty 的属性	114
5.3	getProperty 的属性	115
5.4	forward 的属性	116
5.5	include 的属性	118
5.6	plugin 的属性	120
5.7	实例	122
第六章	JSP 中的字符串处理	129
6.1	构造函数	129
6.2	常见的字符串语法	130
6.3	StringBuffer 类	140
6.4	字符串总结	145
第七章	JSP 与后台数据库管理	146
7.1	关于 JDBC	146
7.2	使用 JDBC	148
第八章	Servlet 编程	208
8.1	关于 Servlet	208
8.2	配置调试环境	209
8.3	如何写作 Servlet	209
8.4	Servlet 的生命周期	210
8.5	编写 Servlet	211
第九章	制作日历	226

第十章 留言板	231
10.1 建立和设置数据库	231
10.2 编写功能层的 JavaBean	237
10.3 编写表示层的 JSP 页面	239
10.4 调试源代码	243
第十一章 网上书店	245
附录 A SQL 语言入门	269
常见的 SQL 语句	269
UPDATE	273
附录 B 常用的 html 标记元素	279
主格式元素	281
字符风格控制元素	282
版面的版式控制元素	284
查询标题及其分级控制元素	286
超文本指针链及锚元素	288
表格元素	289
表单元素	290
附录 C JavaBean	293
附录 D 建立一个聊天室	307

第一章

初 识 JSP

1.1 开发基于 Web 的应用程序：背景

在因特网时代中，由于网站之间的竞争极为激烈，若只使用静态的 `html` 页面则意味着被淘汰。因特网面临着新的挑战，而新的挑战又在于创建交互的基于 `Web` 的应用程序。在这些程序中，页面的内容是基于用户的请求或者系统的状态，而不是预先定义的文字。

最早的解决方案是使用 `CGI`（公共网关接口），在这种解决方案中，每当用户发出一个 `CGI` 要求时，服务器就会启动一个新进程来进行相应的处理。当用户的访问量不大时，这种解决方案还可以应付，但一旦访问量增大，这些进程将消耗该 `Web` 服务器所有的可用资源，并且系统性能急剧下降，从而使访问速度大大降低。

为了在因特网上占有一席之地，各大公司纷纷推出自己的解决方案。某些 `Web` 服务器供应商已经尝试通过为他们的服务器提供“插件”和 `API` 来简化 `Web` 应用程序的开发。这些解决方案往往是与特定的 `Web` 服务器相关的，不能解决跨多个供应商的解决方案的问题。例如，现在比较流行的微软的 `Active Server Pages`（`ASP`）技术使得在 `Web` 页面上创建动态内容更加容易，但是也只能工作在微软的 `IIS` 和 `Personal Web Server` 上。

与前面我们提到的 `ASP` 相比，太阳微系统公司（`Sun Microsystems Inc`）所推出的 `JSP`（`Java Server Pages`）技术基本上解决了这个问题，这是基于 `Java` 语言的跨平台特性。在这里我们需要首先说明的一点是，`JSP` 技术实际上是建立在 `Java Servlet` 技术之上的。`Java Servlet` 是基于 `Java` 技术的运行在服务器端的程序（与 `Applet` 不同，后者运行在浏览器端）。这个 `Servlet` 在服务器端接收来自 `Web` 浏览器的 `HTTP` 请求，动态地生成响应（可能需要查询数据库来完成这项请求），然后发送包含 `html` 或 `XML` 文档

的响应到浏览器上。但是 Java Servlet 的缺点是显而易见的，由于 Java Servlet 的代码是一段 Java 代码，所以当 Web 管理人员想要调整页面显示时，他就需要必须重新编辑并编译该代码。这样，在程序的维护上就有一定的困难；同时，编写 Java Servlet 需要有一定的 Java 水平，这就又为 Java Servlet 的推广带来了一定的困难。广义上的 JSP 技术也包括 Java Servlet，但实际上 JSP 要容易掌握得多，这一点读者在后面的学习中可以体会到。

实际上，JSP 不仅仅只包括网页写作，它覆盖的内容很多，其中主要有：

- (1) 狭义上的 JSP 网页制作；
- (2) Java Servlet 技术。

这两部分对于掌握 JSP 的核心技术来说是必不可少的，我们在后面的章节中将一一讲到它们。

1.2 JSP 的特点

1.2.1 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 JavaBean 组件中，并且捆绑在小脚本中，所有的脚本都在服务器端运行。如果核心逻辑被封装在标识和 Bean 中，那么其他人，如 Web 管理人员和页面设计者，就能够编辑和使用 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 JavaBean 组件、使用 JDBC™ 技术访问数据库，或者包含文件等），并且将结果以 html（或者 XML）页面的形式发送回浏览器。这既有助于作者保护自己的代码，而且又保证了任何基于 html 的 Web 浏览器的完全可用性。

1.2.2 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（JavaBean 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。

1.2.3 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。JSP 技术封装了许多功能，这些功能都是在易用的、与 JSP 相关的 XML 标识中进行动态内容生成所必需的。标准的 JSP 标识能够访问和实例化 JavaBean 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。

通过开发定制化标识库，JSP 技术是可以扩展的。今后，第三方开发人员和其他人员可以为常用功能创建自己的标识库。这使得 Web 页面开发人员能够使用熟悉的工具和如同标识一样的执行特定功能的构件来工作。

JSP 技术很容易整合到多种应用体系结构中，以利用现存的工具和技巧，并且扩展到能够支持企业级的分布式应用。作为采用 Java 技术家族的一部分，以及 Java 2（企业版体系结构）的一个组成部分，JSP 技术能够支持高度复杂的基于 Web 的应用。

由于 JSP 页面的内置脚本语言是基于 Java 编程语言的，而且所有的 JSP 页面都被编译成为 Java Servlet，JSP 页面就具有 Java 技术的所有好处，包括健壮的存储管理和安全性。

作为 Java 平台的一部分，JSP 拥有 Java 编程语言“一次编写，各处运行”的特点。随着越来越多的供应商将 JSP 支持添加到他们的产品中，您可以使用自己所选择的服务器和工具来更改工具或服务器，而不影响当前的应用。

当与 Java 2 平台、企业版（J2EE）和 Enterprise JavaBean 技术整合时，JSP 页面将提供企业级的扩展性和性能，这对于在虚拟企业中部署基于 Web 的应用是必需的。

1.2.4 JSP 页面看上去像什么？

JSP 页面看上去像标准的 html 和 XML 页面，并附带有 JSP 引擎能够处理和抽取的额外元件。通常，JSP 元件创建插入最终页面的文本。

1.3 JSP 页面的应用模型

JSP 页面由 JSP 引擎执行，引擎安装在 Web 服务器或者使用 JSP 的应用服务器上。JSP 引擎接受客户端对 JSP 页面的请求，并且生成 JSP 页面给客户端的响应。

JSP 页面通常被编译成 Java Servlet。后者是一个标准的 Java 扩展，在 www.java.sun.com 站点上对其有更详细的描述。页面开发人员能够访问全部的 Java 应用环境，以利用 Java 技术的扩展性和可移植性。

当 JSP 页面第一次被调用时，如果它还不存在，就会被编译成为一个 Java Servlet 类，并且存储在服务器的内存中。这使得接下来的对该页面的调用有非常快的响应（这

就避免了 CGI-BIN 为每个 HTTP 请求生成一个新的进程的问题，或是服务器端引用所引起的运行时语法分析）。

JSP 页面可以包含在多种不同的应用体系结构或者模型中。JSP 页面可以用于由不同协议、组件和格式所组成的联合体中。

1.4 简单快速地编程

1.4.1 页面作者可以很容易地根据请求和资源状态生成动态内容

这个结构在许多应用上工作良好，但不能扩展到大量的基于 Web 的并发客户访问稀少的企业资源，因为每个客户必须建立或者共享一个到可用内容资源的连接。例如，如果 JSP 页面访问数据库，则可能生成许多到数据库的连接，这将影响到数据库的性能。

1.4.2 使用 Java Servlet 的一个灵活的应用

在另一种可能的配置中，基于 Web 的客户机可能直接对 Java Servlet 进行请求，Servlet 生成动态内容，将结果捆绑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（例如 html）发送回浏览器。

JSP 页面也可以作为企业级 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。

1.5 JSP 页面与 XML 技术的集成

JSP 页面可以用于生成 XML 和 html 页面。

对于简单的 XML 生成，可以包含 XML 标识和 JSP 页面的静态模板部分。对于动态 XML 生成，则使用基于服务器的对象和生成 XML 输出的客户化标识。

JSP 页面与 XML 工具不是不兼容的。尽管 Sun 在设计 JSP 规范时使得 JSP 页面即使手工操作对于创作者而言也是很容易的，JSP 规范同时也提供了一个机制以便于创建任意 JSP 页面的一个 XML 版本。通过这种方法，XML 工具能够创作和操作 JSP 页面。

通过将 JSP 标识和元件转换为与 XML 兼容的对等物，可以使用基于 XML 的工具

来操作 JSP 页面。例如，脚本可以被包含在<%和%>之中，或者基于 XML 标志的<jsp:scriptlet>和</jsp:scriptlet>中。事实上，经过下面这样简单的几步将 JSP 页面转换为 XML 页面是可能的，这些步骤包括：

- 增加一个 JSP 根元件
- 将元件和指示转换为 XML 兼容的对等物
- 为页面上其他的元件（通常为非 JSP）创建 CDATA 元件

通过这个与 XML 兼容的方法，创建 html 页面的设计者仍然拥有一个快速创建动态 Web 页面的易用环境，同时，基于 XML 的工具和服务可以与 JSP 页面集成并且和 JSP 兼容的服务器一起工作。

1.6 JSP 技术的未来

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。

通过与业界领袖的合作，Sun 保证 JSP 规范是开放和可移植的。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来扩展平台的功能。

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 JSP1.0 的支持，这样就在业界已经拥有了最初的、立即的支持。

1999 年早些时候完成的 1.1 版本，可通过更多的 XML 支持、客户化标识、以及与 J2EE 的集成来得到扩展。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。

第二章

安装 JSP 的使用环境

在我们走进奇妙的 JSP 世界之前，我们需要一个 JSP 测试环境。由于 JSP 是一种以 Java 为主的跨平台语言，因此我们可以在许多操作系统上建立自己的 JSP 测试环境，这也是 JSP 的优点之一。Sun 公司免费提供各种开发工具来供 Windows、Solaris，以及 Linux 平台使用。

2.1 在 Windows 下的安装与设置

本书的所有源代码都是在 Windows 98+jsdwk 中进行调试的，所以本书将详细介绍 Windows 98 下的安装。Windows NT 下的安装与此类似。

在我们进行安装之前，我们需要安装的组件有：

(1) Java 工具包 JDK1.2.2。

该工具包可以从 <http://java.sun.com/jdk/> 上下载。

(2) JavaServer 网站发展工具 JSWDK (JavaServer Web Development Kit 1.0.1)。

该工具可以从 <http://java.sun.com/products/jsp/> 上下载。

安装过程：

第一步：安装 Java 工具包 JDK1.2.2

(1) 双击下载的 jdk1_2_2-win.exe 文件进行安装，使用默认配置进行安装，JDK 的默认安装目录为 C:\jdk1.2.2，JRE 的默认安装目录为 C:\Program Files\JavaSoft\JRE\1.2。

(2) 重启计算机。

第二步：安装 JSDKW

将下载的 jsdkw1.0.1-win.zip 文件进行解压，假设 zip 文件被解压到目录 D:\JSP 中，则我们会看到在目录 D:\JSP 下出现了目录 jsdkw-1.0.1。该目录中所包含的文件已经列出，如图 2-1 所示。

名称	大小	类型	修改时间
etc		文件夹	2001-2-22 20:04
examples		文件夹	2001-2-22 20:04
lib		文件夹	2001-2-22 20:04
src		文件夹	2001-2-22 20:04
webpages		文件夹	2001-2-22 20:04
FAQ.html	9 KB	HTML Document	1999-9-23 8:29
LICENSE	9 KB	文件	1999-9-23 8:29
README.html	7 KB	HTML Document	1999-9-23 8:29
startserver	2 KB	文件	1999-9-23 7:30
startserver.bat	2 KB	MS-DOS 批处理文件	1999-9-23 7:30
stopserver	2 KB	文件	1999-9-23 7:30
stopserver.bat	1 KB	MS-DOS 批处理文件	1999-9-23 7:30
webserver.jar	115 KB	Executable Jar File	1999-9-23 7:58
webserver.xml	8 KB	XML Document	1999-9-23 7:30

图 2-1 安装 JSDKW 后的文件列表

第三步：更新下列环境变量

(1) 在开始菜单中单击“运行”，然后在出现的对话框中键入“sysedit”，则系统会弹出一个如图 2-2 所示的窗口。

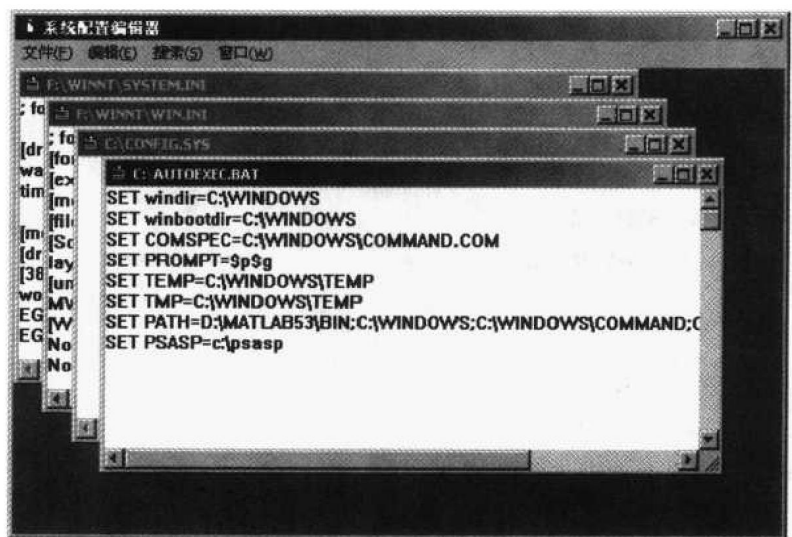


图 2-2 系统配置编辑器

- (2) 在 autoexec.bat 文件中把 C:\jdk1.2.2\bin 目录追加到 PATH 中。
- (3) 在 autoexec.bat 中写入 SET JAVA_HOME=C:\jdk1.2.2。
- (4) 在 autoexec.bat 中写入 SET CLASSPATH=C:\jdk1.2.2\lib\tools.jar, 并保存文件。
- (5) 重启计算机。

第四步：设置 JSP 测试环境

在 JSDWK 中，其测试环境的参数都在 XML 文件 webServer 中，用文本编辑器打开后该文件的正文内容如下：

```
<!--ATTLIST WebServer
    id ID #REQUIRED
    adminPort NMTOKEN "">

<!--ELEMENT Service (WebApplication*)>
<!--ATTLIST Service
    id ID #REQUIRED
    port NMTOKEN "8080"
    hostName NMTOKEN ""
    inet NMTOKEN ""
    docBase CDATA "webpages"
    workDir CDATA "work"
    workDirIsPersistent (false | true) "false">

<!--ELEMENT WebApplication EMPTY>
<!--ATTLIST WebApplication
    id ID #REQUIRED
    mapping CDATA.#REQUIRED
    docBase CDATA #REQUIRED
    maxInactiveInterval NMTOKEN "30">
]>

<WebServer id="webServer">
    <Service id="service0">
        <WebApplication id="examples" mapping="/examples" docBase="examples"/>
    </Service>
</WebServer>
```

其中有几个比较重要的参数:

docBase: 表示放 JSP 主页的根目录, 默认值为 jsdkw 目录中的 webpages 目录。

workDir: 根据第一章中的内容, 我们知道, JSP 引擎将 JSP 主页翻译为 servlet 后再将 servlet 编译为机器代码并在服务器端运行, workdir 参数表示的就是存放编译后的机器代码的目录, 默认的目录为 jsdkw 中的 work 目录。

port: servlet 运行的端口号, 默认值为 8080, 你也可以设该值为其他数字, 比如“9090”。

第五步: 启动 JSP 测试环境

(1) 进入 D:\JSP\jsdkw-1.0.1 目录, 双击 startserver.bat, 则出现一个 DOS 窗口, 如图 2-3 所示。

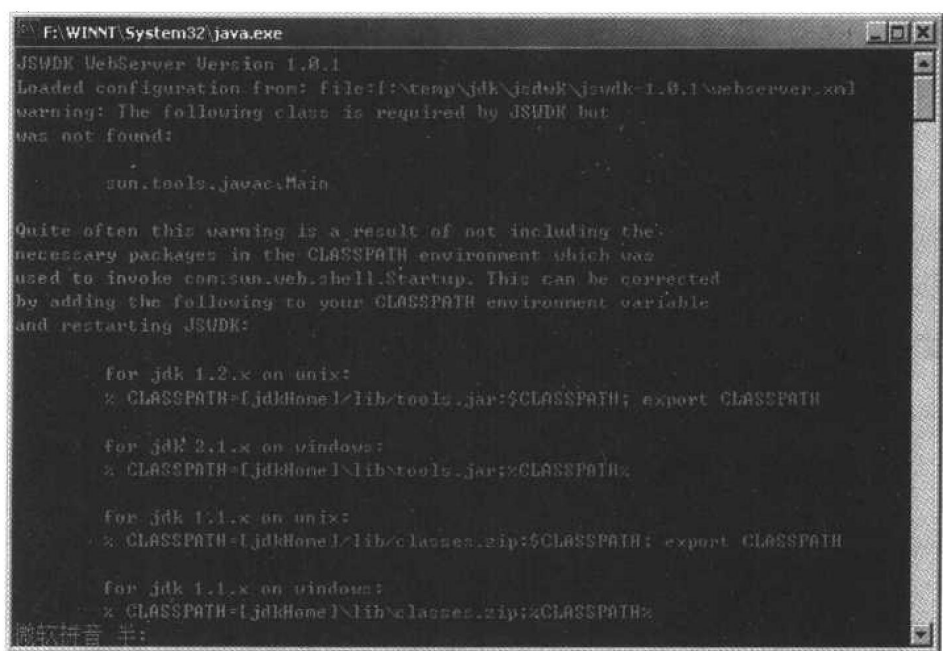


图 2-3 启动 JSP 服务器

(2) 启动网页浏览器, 在其地址栏里键入 <http://127.0.0.1:8080>。如果出现图 2-4 所示内容, 那么就说明我们已经成功地配置好了 JSP 测试环境, 以后就可以放心地写 JSP 页面了。

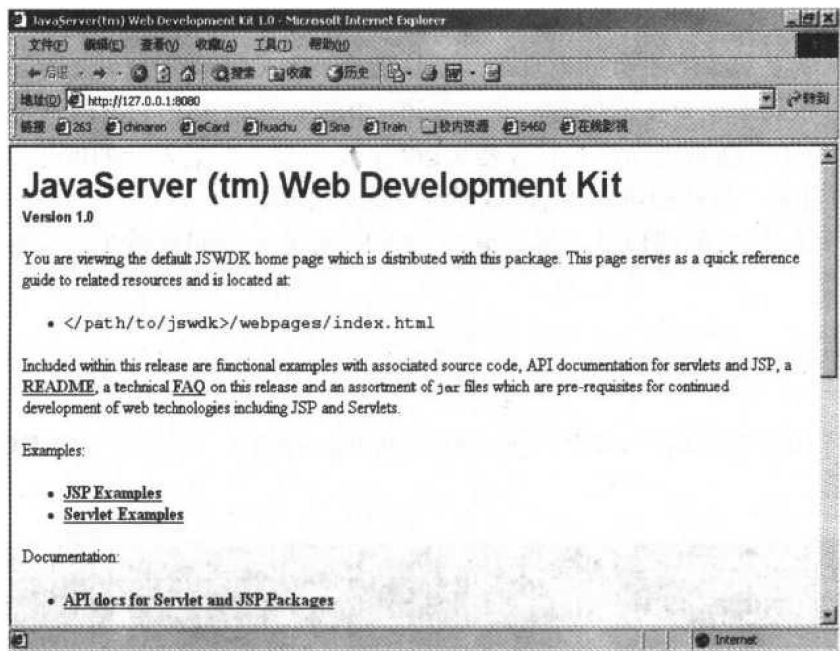


图 2-4 配置成功后的页面

配置过程中可能会遇到一些问题，下面我们针对几个常见问题做个讨论，并给出其解决方法。

问题之一：

单击 `startserver.bat`，则系统会弹出如图 2-5 所示的 DOS 窗口。

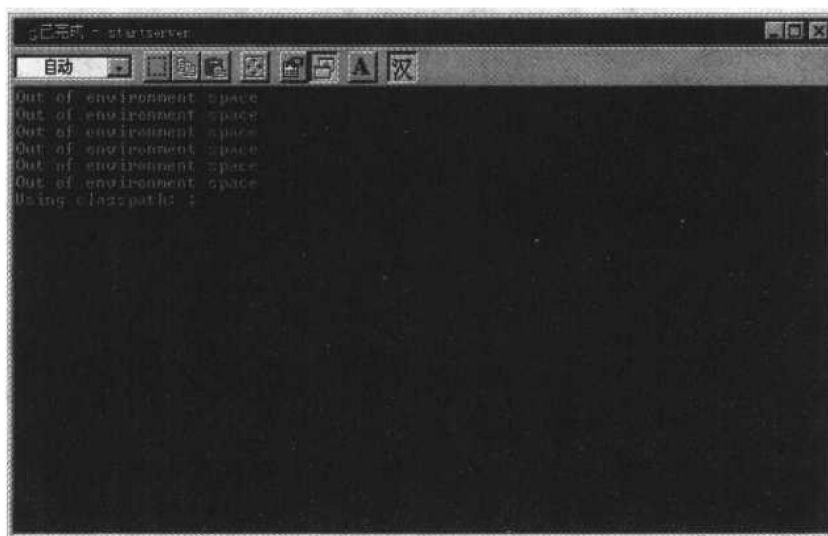


图 2-5 内存不足提示窗口

这表示系统设置的内存不足。解决办法如下：