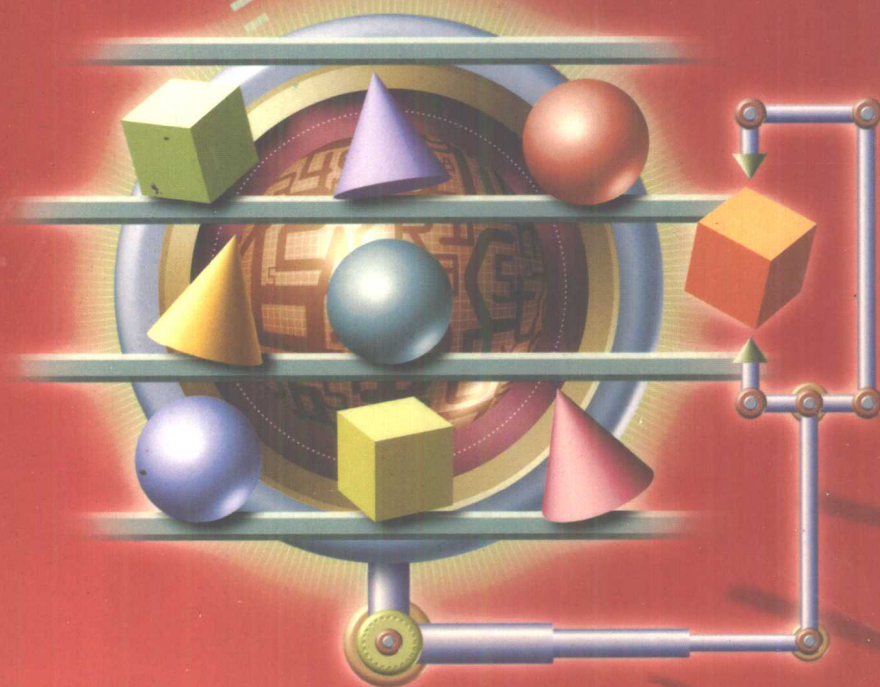


JSP 即时应用



Paul Tremblett 著
高波 王琰 易粟 等译

人民邮电出版社
www.pptph.com.cn

- 50余个即用解决方案
- 通过主题组织易于参考
- 涵盖公司Intranet、电子商务站点及XML的JSP应用

麦格劳-希尔国际公司
www.mhhe.com

Mc
Graw
Hill

JSP 即时应用

Paul Tremblett 著

高 波 王 琰 易 粟 等译

 人民邮电出版社 麦格劳-希尔国际公司 

内 容 提 要

本书介绍了如何用 JavaServer Pages(JSP)创建强大的、动态的基于 Web 解决方案,内容实用、富有指导性。本书第 1 章介绍了 JavaServer Pages 技术并举了一些简单的例子。第 2 章讲述 JSP 的元素。余下部分讲述如何使用 JSP 来解决各种普通的实际问题,其中的实例覆盖了 JSP 基本内容,包括为公司的 Intranet 和电子商务的 Web 站点创建大量的 Web 应用。附录 A 和附录 B 帮助读者创建一个运行范例程序的环境。附录 C 讲述 JSP API。因为范例代码使用了某些 Servlet API,所以与这些 API 的相关的部分在附录 D 中进行了说明。附录 E 列出了随书所附光盘的内容。

本书的主要读者对象为网络工程师、Web 开发人员及大专院校师生。

JSP 即时应用

- ◆ 著 Paul Tremblett
- 译 高 波 王 琰 易 粟 等
- 责任编辑 梁 凝
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@puph.com.cn
网址 <http://www.puph.com.cn>
北京汉魂图文设计有限公司制作
北京朝阳展望印刷厂印刷
新华书店总店北京发行所经销
- ◆ 开本:787×1092 1/16
印张:29.25
字数:736 千字 2001 年 1 月第 1 版
印数:1-5 000 册 2001 年 1 月北京第 1 次印刷
著作权合同登记 图字:01-2000-3964 号

ISBN 7-115-08925-6/TP·1921

定价:57.00 元(附光盘)

版 权 声 明

Paul Tremblett: Instant JavaServer Pages

ISBN 0-07-212601-9

Copyright © 2000 by the McGraw-Hill Companies, Inc.

Authorized translation from the English language edition published by McGraw-Hill, Inc.

All rights reserved.

本书中文简体字版由人民邮电出版社和美国麦格劳-希尔国际公司合作出版。未经出版者书面许可，不得以任何方式复制或抄袭本书的任何部分。

版权所有，翻印必究。

致 谢

谨以此书献给我亲爱的妻子 Eleanor。是她使我从没有自信和勇气的状态开始走上了一个精彩的征途，是她在我前进的每一步中给了我鼓舞和支持。

感谢我的父母，是他们使我喜爱上了阅读。那写满字母表的被子为我开启的知识之门是超乎于想象的。

我还必须感谢我的妻子，当我写这本书时，她承受了不同寻常的孤独。只是字典中没有足够的词汇来表达清楚，Ellie，就让我说声谢谢吧。

最后，我还想对 Adrian Colyer 致以特殊的谢意。他的那些来自大西洋彼岸的提示、指导、对代码的更正以及建设性的评论，都对本书作出了重要的贡献。

——Paul Tremblett

前 言

在短短的几年中，Java 已经成为了基于 Web 应用的语言选择。JavaServer Pages 技术填补了服务器端 Java 的强大性与浏览器端 HTML 灵活性之间的差距。填补这种差距的方法是允许内容和表达之间的分离。这种分离使得 Web 设计者和程序员可以共同工作并进而达到一个共同目标，而且在这个过程中不会相互妨碍对方的工作，也不会产生相互不利的依赖性。

世界并不是由 Web 服务器和浏览器来定义的，JSP 并不将自己局限于 HTML。它能同样满意地用来在企业应用组件之间传递 XML，并将 WML（无线标记语言）传给无线设备。

本书的目的就是介绍 JavaServer Pages 技术并帮助读者尽快地开始使用。

本书的结构

本书第 1 章介绍了 JavaServer Pages 技术，并举了一些简单的例子。第 2 章讲述 JSP 的元素。余下部分讲述如何使用 JSP 来解决各种普通的现实世界的问题。如果安装、执行并学习每个范例程序，你将使用到 JSP 版本 1.0 规范中的所有元素。

附录 A 和附录 B 帮助读者创建一个运行范例程序的环境。附录 C 讲述 JSP API。因为范例代码使用了某些 Servlet API，所以与这些 API 相关的部分在附录 D 中进行了说明。附录 E 列出了随书所附光盘的内容。

本书面向的读者

本书是为那些善于实践的程序员设计的。JSP 元素的语法以及每个元素的功能在前面的两章中讲述。剩下的部分全部用于讲述程序员每天的工作——编程。

关于范例程序

独立的代码“片断”可能有助于表述语法，但不如完整的应用程序更有价值。所以本书采用了后者。无可否认，范例程序并不是最好的设计典范，它们的目的是介绍使用 JSP 的方法。如果当你运行范例程序后，能有这样的想法：“OK，我知道这是怎么做的，但我能使用其他方法”，那么这个范例就达到了它本来的目的。

这里关于没有错误的代码定义是：导致错误的原因还没有被发现的代码。几乎没有会发生错误的可能。光盘中的代码也可从网址 www.instantjsp.com 上下载。一旦程序中有错误被发现，该网址上的程序代码将反映程序更新的情况。

对未来的展望

任何期望成功的技术，都必须不断完善与发展。JavaServer Pages 技术正是如此。本书使用 JSP 规范的 1.0 版本。1.1 版本的规范已经出版，作为 Java 2 企业版的一个完整部分，JSP 又有了新的任务。该规范可在网址 java.sun.com/products/jsp/download.html 上获得。如果

对该规范进行调查，将会发现 JSP 现在能够支持完全与企业的类应用相集成的、复杂的 Web 站点。

Paul Tremblett

目 录

第 1 章	JavaServer Pages 介绍	1
1.1	什么是 JavaServer Pages	2
1.2	如何访问 JavaServer Pages	3
1.3	使用 JSP 技术的好处	3
1.4	何时使用 JavaServer Pages	4
1.5	JSP 和 ASP 的比较	4
1.6	一些范例	5
1.6.1	必不可少的“Hello World”	5
1.6.2	有谁真正知道现在几点吗	6
1.6.3	现在得到你的信息	8
1.6.4	从请求中提取信息	24
第 2 章	JSP——更详细的介绍	27
2.1	指令	28
2.1.1	page 指令	28
2.1.2	include 指令	36
2.1.3	taglib 指令	37
2.2	动作	37
2.3	声明	55
2.4	Scriptlets	56
2.5	表达式	56
2.6	jspInit 和 jspDestroy 方法	57
2.7	对所有 JSP 可用的对象	58
2.8	JSP 页创建的对象	60
第 3 章	实现一个典型的 Web 会话	61
3.1	登录方案	62
3.2	对 LoginManager 的进一步探讨	68
3.3	登录	70
3.4	防止用户绕开登录	76
3.5	改善的登录过程	83
3.6	限制非法登录企图	88
3.7	捕获及处理休止状态	93

3.8	测量整个会话时间	95
3.9	监控和配置活动	97
3.10	注销	99
3.11	SessionManager 代码	102
第 4 章	在企业内部网中使用 JSP	111
4.1	创建一个员工数据库	112
4.2	允许员工访问他们的记录	112
4.2.1	查看谁在我的部门内	122
4.2.2	检查假期情况	132
4.2.3	查看我的下一次验收时间	135
4.3	员工电话簿	136
4.4	公司的技能数据库	140
4.5	创建销售资料知识库	145
4.6	使销售人员更轻松	148
4.7	创建知识库	153
4.8	使用知识库	153
第 5 章	在线商店的第一步——显示商品	163
5.1	创建产品数据库	164
5.2	在线商店	167
5.3	浏览在线目录	169
5.4	搜索在线目录	174
5.5	跟踪顾客的兴趣	180
5.6	StoreBean 的代码	204
5.7	Customer Bean 的代码	210
第 6 章	网上销售	219
6.1	网上订购	220
6.2	订单追踪及顾客维系	242
6.3	购物单的源代码	247
第 7 章	如何听到顾客的意见	249
7.1	顾客反馈表	250
7.2	顾客满意表单	255
7.3	产品评价表	265
7.4	操作随机顾客调查表	274
7.5	CustomerInput 的源代码	285
7.6	Survey Bean 的源代码	288

第 8 章 奖励和特别派送	291
8.1 常客程序	292
8.2 特殊赠送	295
8.3 公共信息	297
8.4 “特殊赠送” Bean	297
8.5 PreferredCustomerNotices.jsp	300
第 9 章 一些有趣的 JSP	303
9.1 一个传统的趣味游戏——“运气”小甜饼	304
9.2 投票支持你喜欢的歌	308
9.3 发送电子明信片	322
9.4 一个简单的聊天室	335
第 10 章 超越 HTML	351
10.1 使用 XML	352
10.2 检索 PostScript 文件	363
附录 A 创建一个 JSP 环境	367
A.1 操作系统	368
A.2 JSP 环境	368
A.3 数据库	388
A.4 Apache 安装	388
A.4.1 简介	388
A.4.2 通过 APACI 安装 Apache 1.3 HTTP 服务器	389
A.5 在 UNIX 上安装 Apache JServ	397
A.5.1 简介	397
A.5.2 安装 Apache JServ Servlet 引擎	397
A.6 安装 PostgreSQL	400
附录 B 设置 PostgreSQL 数据库	405
B.1 配置和运行 PostgreSQL	406
B.2 数据库	406
B.3 数据表	406
附录 C JSP 应用程序接口 (API)	409
C.1 javax.servlet.jsp HttpJspPage 接口	410
C.2 javax.serlet.jsp JspEngineInfo	410
C.3 javax.serlet.jsp Class JspFactory	411

C.4	javax.servlet.jsp JspPage 接口	413
C.5	java.servlet.jsp JspWriter	414
C.6	javax.servlet.jsp PageContext	422
附录 D	Servlet 应用程序接口	431
D.1	javax.servlet.http.Cookie Cookie	432
D.2	javax.servlet.http.HttpServletRequest Request HttpServletRequest javax.servlet.http. HttpServletRequest	436
D.3	javax.servlet.http.HttpServletResponse Response HttpServletResponse javax.servlet.http. HttpServlet	440
D.4	javax.servlet.http.Session HttpSession javax.servlet.http.Session	446
D.5	javax.servlet.http.HttpUtils HttpUtils javax.servlet.http.HttpUtils	450
D.6	javax.servlet.ServletException ServletRequest javax.servlet.ServletException	451
D.7	javax.servlet.ServletResponse ServletResponse javax.servlet.ServletResponse	455
附录 E	光盘中的内容	457

第 **1** 章

JavaServer Pages 介绍

- 日期和时间显示
- 复选框和单选钮
- 列表框

在本章中，我们将知道什么是 JavaServer Pages (JSP)TM以及它们是如何工作的。我们能够认识使用 JavaServer Pages 的好处，并将 JSP 与 ASP (Active Server Pages) 相比较。通过检查一些范例程序以及它们产生的结果，我们能学习某些简单的 JSP 元素。

1.1 什么是 JavaServer Pages

在 Web 流行的早期，对访问一个典型站点时浏览的页面最佳的形容就是“电子小册子”。这些网站的内容是静态的，访问者一旦看过某个网站后，很少会愿意再次访问该站点。

因此，什么样的网站才能让人访问不止一次呢？如果你所拥有的股票在昨天的情形不是很好，而今天你又希望看到它的走势，那么发布接近于实时股票报价的站点就是你希望再次光顾的站点。如果你的即将到来的加勒比海旅行好像要被飓风所阻碍，那么显示当前和预计的风暴情况的网站将会使你在出发之前访问多次。如果你正注意到一台超出你预算的打印机，你将会对一个经常提供打折服务的网站至少每星期访问一次。这些网站吸引你回头光顾的共同之处是什么？答案就是反映现实情况的动态内容。

最早试图产生动态内容的是公共网关接口 (CGI)。它依赖于运行在服务器端的 Perl 脚本或 C 程序。这些语言的“写”语句——Perl 语言的 `print` 和 C 语言的 `printf`——使动态生成 HTML 成为可能，但并不一定使 Web 设计者的工作变得更容易。因为 Web 设计者并不一定是程序员。一个解决的办法是让 Web 设计者创建包含特殊标记的 HTML 模板，动态内容必须插入在这些标记处。Perl 或 C 代码将读取并解析这些模板文件，在传送 HTML 到浏览器之前在指令的地方执行替代内容。这样产生的问题是当读模板文件时会产生性能损失。读这些文件导致性能降低是因为每收到一个请求，就得加载 Perl 解释程序或 C 程序。CGI 的另一个问题是它的状态是不定的。会话的效率通常是由跟踪在客户和服务端之间来回传输的隐藏域的内容来实现的。会话管理的责任压在了程序员的身上，有多少程序员，就几乎有多少种对会话管理的解决方法。CGI 的第三个或许是最严重的问题就是业务逻辑被包含在形成表示层的代码中。一方的改变可能会影响到另一方。

Java servlet 解决了 CGI 的两个问题。第一个是性能问题。servlet 被装载一次，要么在服务器启动时，要么当第一个对 servlet 的请求到达服务器，或某些情况下当新版的 servlet 可用时。第二个是会话管理问题。servlet 容器为每个用户自动生成一个唯一的标识符。它还能提供一种将此标识符和会话对象相联系的方法。在这个对象中，特定应用程序的数据被保存并跟踪这些标识符和关联。servlet 程序员能够集中精力于逻辑和表达，使用明确定义的 API 来访问会话服务。servlet 动态生成 HTML 的实现采用同样的方法——使用“写”语句。所有的都是使用 `print()` 和 `println()` 方法。表示层程序代码中。

表示层的分离由 JavaServer Pages 来实现。JSP 页是一个基于文本的文档，它描述如何处理请求和产生回应。该描述将固定的模板数据与一些动态行为混合起来。JavaServer Pages 规范是建立在 Java servlet 规范提供的框架基础上的。支持 JavaServer Pages 的语义模型也就是 servlet 的语义模型。最常用的实现包括一个只执行一次的转换阶段，以及一个每次请求就执行一次的请求处理阶段。转换阶段通常使用文本文档的内容来创建一个实现

javax.servlet.Servlet 接口的类，这和 servlet 的周期是一样的。JSP 容器作为一个组件被安装在 Web 服务器上或启用 Web 的应用程序服务器上，容器将从客户端接收到的请求传送给源于 JSP 页的类中的 `_jspService()` 方法。这个方法是源 JSP 文档中指定的动态行为的编译形式，它准备一个应答，通过 JSP 容器传送到客户端。

我们将要学习的例子会清楚地说明 JavaServer Pages 最常用于生成 HTML。然而这并不能推断出 JavaServer Pages 只局限于生成 HTML。JavaServer Pages 能产生任意文本的输出。例如，在企业对企业的环境下，JavaServer Pages 能用于产生 XML 文档。在无线上网变得越来越流行的情况下，JavaServer Pages 甚至可以用来生成无线标记语言（WML），来实现手提设备如移动电话之间的通信。

1.2 如何访问 JavaServer Pages

我们已经知道如何访问静态 HTML 页面，即只需要敲入一个 URL 地址，如 <http://dormouse.instantjsp.com/SimplePage.html>。浏览器端使用超文本传输协议（HTTP）来向运行于 instantjsp.com 域 dormouse 主机上的 Web 服务器发送一个对 SimplePage.html 内容的请求。Web 服务器取出文件传回浏览器端，并在文件中使用 HTML 标记来直观呈递页面。

客户端要求访问 JavaServer Pages 页面的方法也是同样的，只是指定文件扩展名为 .jsp 而不是 .html，因此前面提到的 URL 地址就成为 <http://dormouse.instantjsp.com/SimplePage.jsp>。请求被传送到支持 JSP 的 Web 服务器或启用 Web 的应用程序服务器，服务器能识别 .jsp 扩展名并将请求传递到 JSP 容器，容器调用已编译的 JSP 页的 `_jspService()` 方法。`_jspService()` 方法产生的应答被送回到 JSP 容器，然后送回至客户端。注意，这并不是说对 JavaServer Pages 的请求一定产生于浏览器端。大多数情况下是如此，但它也能从 servlet 或另一个 JSP 页产生。

注意：虽然“JSP 页”这个术语有点重复，但它被开发者广泛运用，所以在本书中也采用此说法。

1.3 使用 JSP 技术的好处

JSP 页最重要的好处是将程序与界面相分离的能力。在使用 Web 的早期甚至是现在，很大程度上一个困扰 Web 站点的主要问题是：艺术家制作的页面往往很吸引人，但却缺乏实用性。另一方面，程序员制作的页面功能虽强，但是缺乏吸引访问者所必需的视觉效果。程序员与美工设计者协作完成的页面解决了这个问题，但是事实证实了当维护的时候又会产生一些问题。美工设计者经常发现要改变一个网站的外表而不涉及到程序员是不可能的。同样地，程序员有时会发现程序的改变影响到了视觉效果。如果使用 JSP，双方就能够真正独立地工作。

如 Java 所宣称的一样，JSP 的另一个重要的好处可简单概括为：一次编写，各处运行（Write Once, Run Anywhere™）。通过最终转换为 Java 字节代码这样的优势，JSP 页是独立

于平台的，这意味着 JSP 能够在任何平台上开发，在任何服务器上使用。

JSP 页也使植入可复用组件变得简单。组件（例如 JavaBean）和执行特殊任务的 Enterprise JavaBean 能被只开发一次就组合到任意数量的 JavaServer Pages 页中。想像一下如果你被分配的任务是开发一个 Web 应用，主要目的是为一遗产体系提供一个现代化的用户界面。你可能会花相当多的时间来评估商业的 Enterprise JavaBeans，来使你的 JSP 页与 CICS 进行通信，或执行 3270 终端仿真，或者你决定自己来开发。如果你下一个任务是为另一个遗产体系做同样的工作，你只需在你的 JSP 页中包含那些你已开发的 Enterprise JavaBeans 所需要的指令。除了可复用的组件外，JSP 页还能够静态或动态地包含其他 JSP 代码。

JavaServer Pages 技术现在是 Java 2 Enterprise Edition (J2EE) 的一个完整的部分，J2EE 给企业计算带来了 Java 技术。通过使用 JSP 页来创建 Web 站点，能创建一个强大的 N 层应用类型的前端组件。J2EE 使这种应用成为可能。

JSP 的另一个好处是自定义标记库带来的可扩展性。除了标准的必须被所有一致 JSP 容器实现的动作类型，JSP 规范版本 1.0 提供了一个 Taglib 指令来用于引进新的动作类型。这种方法简化了 JSP 容器开发者的工作，减少了 JSP 创作者的学习时间。规范预言，以后版本的 JSP 规范将包含一个标记扩展结构来描述所有非标准动作的语义。JavaServer Pages 规范的 1.1 版本提供了更多的关于 Taglib 指令是如何被实现的细节。Taglib 支持保证能在所有 JSP 容器间工作——它能被动态装载，不依赖于提供 JSP 容器方的原来的支持。

标记库中定义的 JSP 元素在设计时以页面构成工具中的控件出现。标记库还能被 JSP 写作工具使用。

1.4 何时使用 JavaServer Pages

前面已经讲到 JSP 页有利于业务逻辑与界面的分离。在每次决定是写 servlet 还是使用 JSP 页的时候考虑这点是明智的。在了解了 JavaServer Pages 的强大功能之后，你将会很容易被以下观点所吸引：既然 JSP 页总会被转换并编译成实现 javax.servlet.Servlet 的类，那么为什么不在所有的情况下都使用 JSP 呢？如果这样，那么你又将回到了最初的起点，业务逻辑和界面效果在共同的代码中实现。如果还记得 servlet 适于业务逻辑，JSP 适于表达，你将作出正确的设计决定。

1.5 JSP 和 ASP 的比较

JavaServer Pages 不是最好的，当然也不是唯一的服务器端编程技术。一个应用得更广泛的编程技术是微软的 Active Server Page (ASP)。JSP 和 ASP 都可替代 CGI，而且它们的共同之处在于最终产品都是动态生成的 HTML；然而，它们之间存在着显著的差异。表 1-1 表示了 JSP 和 ASP 的比较。

表 1-1 比较 JSP 和 ASP

	JSP	ASP
平台	Java 被移植的任何平台（Solaris, Linux, Windows NT, Mac OS, AIX, HP-UX 以及各种其他 UNIX 形式）	Microsoft Windows（第三方移植产品可在某些其他平台上）
Web 服务器	任何 Web 服务器（比较流行的包括 Apache, Netscape, IIS）	Microsoft IIS 和个人 Web 服务器
编程语言	Java	VBScript, JavaScript
可复用, 交叉平台组件	JavaBeans, Enterprise JavaBeans	无
内存遗失保护	是	否
可自定义标记	是	否
数据库集成	任何支持 JDBC 或 ODBC 技术的数据库	任何使用 ODBC 的数据库
组件	JavaBeans, Enterprise JavaBeans, 可扩展 JSP 标记	COM

1.6 一些范例

范例程序代码是一个有效的学习工具，它在本书中被广泛运用。每个范例包含了一个或多个 JSP 页，有时候是 HTML。在合适的地方，本书还给出了 Java 类的源代码。范例运行时浏览器端页面的显示也一并给出。

所有的范例都是依照 JavaServer Pages 规范 1.0 版本来写的。附录 A 指导如何装载和配置运行范例程序所需的服务器软件，可以选用任意浏览器。

1.6.1 必不可少的“Hello World”

当今每种计算机语言的学习几乎都从这个无处不在的“Hello World”开始。这里是显示这个著名信息的 JavaServer Pages 页的代码：

```
<html>
<head>
<title>Hello World</title>
</head>
<body bgcolor=" #FFFFFF" >
<font face = " Arial, Helvetica" ><font size=" +2" >
<br>
<br>
<br>
<center>
<b>
```



```
Hello World!  
<p>  
I'm ready for some Instant JSP!!!!  
</b>  
</center>  
</font>  
</body>  
</html>
```

图 1-1 说明了怎样利用浏览器请求该 JSP 页。当页面执行时,会产生如图 1-2 所示的输出。现在,你将无法找出这个 `Hello.jsp` 和你最初学习 HTML 时写的 `HelloWorld.html` 之间内容上的差异。那是因为 JSP 页中所有不属于 JSP 元素的文本都被 JSP 容器处理为固定的模板数据,并被一字不差地传送到客户端。

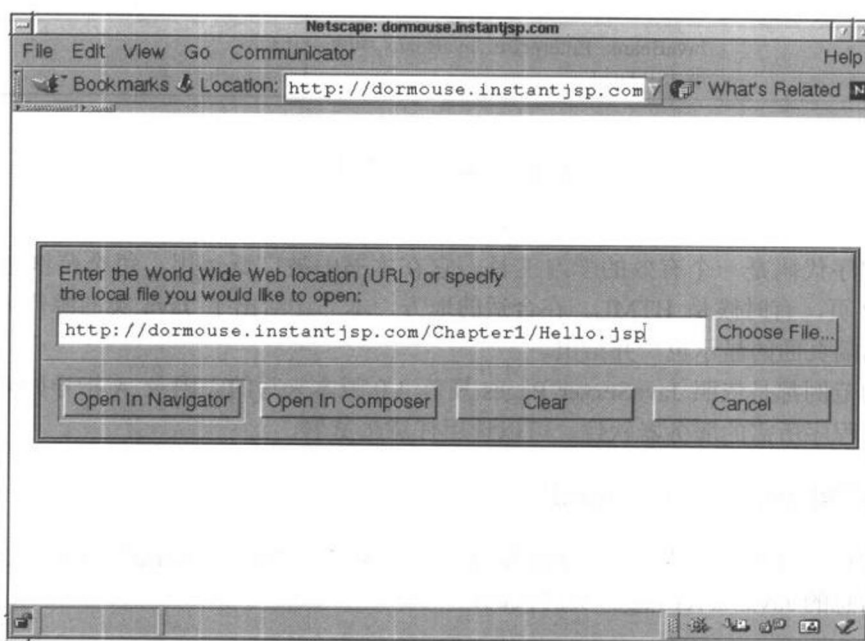


图 1-1 请求 JSP 页

1.6.2 有谁真正知道现在几点吗

现在我们来看一个超出简单 HTML 范围的范例。更重要的是它能产生动态输出。这个例子显示请求页面时的日期和时间。下面是代码:

```
<html>  
<head>  
<title>What Time Is It?</title>  
</head>
```