清华电脑学堂

JSP

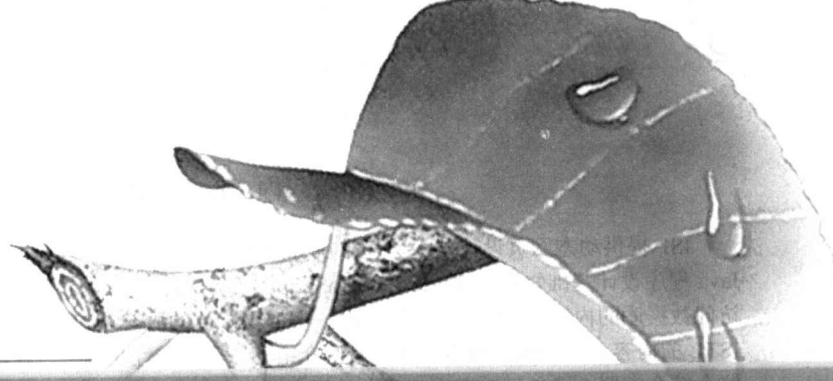
动态网站开发 实践教程

张银鹤 刘治国 张豪 等编著

- 总结了作者长期教学培训成果，难易适中，实用性强
- 系统全面地介绍了JSP的技术要点
- 围绕丰富的实例讲解动态网站开发实践知识
- 精心编写了大量“实验指导”，引导学生深入学习动态网站开发实践
- 网站提供代码下载和课件支持

清华大学出版社

清华电脑学堂



JSP 动态网站开发 实践教程

张银鹤 刘治国 张豪 等编著

清华大学出版社
北京

内 容 简 介

JSP 提供动态网页设计技术，建立在 Java Servlet 技术基础之上。本书全面介绍了 JSP 基础知识，Java 程序设计基础知识，HTML 和 JavaScript 的内容，JSP 中的各种页面元素，JSP 的内置对象， JSP 访问数据库的内容，JSP 中的组件技术——JavaBean 的内容，JSP 中使用 XML 的技术，自定义标记库 JSTL 的内容，Servlet 的内容。本书最后介绍了一个综合性实例——在线图书商城的制作全过程。通过这个综合实例，读者可以在实际开发过程中深入理解 JSP 的内容。

本书可以作为读者学习 JSP 和动态网站开发的教程，适合于普通高校计算机专业和非计算机专业的动态网站开发教程，也可供自学读者使用。

图书在版编目（CIP）数据

JSP 动态网站开发实践教程/张银鹤，刘治国，张豪等编著. —北京：清华大学出版社，2007.1
ISBN 978-7-302-14339-0

Ⅰ. J…　Ⅱ. ①张…　②刘…　③张…　Ⅲ. JAVA 语言－主页制作－程序设计－教材
Ⅳ. TP393.092

中国版本图书馆 CIP 数据核字（2006）第 154722 号

责任编辑：夏兆彦　刘　霞
责任校对：张　剑
责任印制：杜　波
出版发行：清华大学出版社　　**地　　址**：北京清华大学学研大厦 A 座
http://www.tup.com.cn　**邮　　编**：100084
c－service@tup.tsinghua.edu.cn
社 总 机：010-62770175　**邮购热线**：010-62786544
投稿咨询：010-62772015　**客户服务**：010-62776969
印 装 者：北京鑫海金澳胶印有限公司
经　　销：全国新华书店
开　　本：185×260　**印　张**：26.25　**字　数**：620 千字
版　　次：2007 年 1 月第 1 版　　**印　　次**：2007 年 1 月第1 次印刷
印　　数：1～5000
定　　价：39.00 元

前言

FOREWORD

Java Server Pages（JSP）是基于 Java 语言的服务器端脚本技术。JSP 提供动态网页设计技术，建立在 Java Servlet 技术基础之上。通过 JSP 技术，程序开发人员可以利用 JSP 的标记在 HTML 静态内容中加入动态数据，制作动态网页。HTML 代码的编写可以由网页设计人员负责，而动态数据则由程序开发人员负责，这样可更有利于网页设计者和程序开发人员发挥各自的特长。

本书内容

本书介绍了 JSP 的基础知识，包括 Web 的演变、JSP 的工作原理和执行过程以及配置 JSP 服务器等；Java 程序设计的基础知识，这是掌握 JSP 的必备基础；制作网页的基础——HTML 和 JavaScript 的内容。

本书全面介绍了 JSP 中的各种页面元素，包括 JSP 指令、脚本和动作元素等；JSP 的内置对象，包括输入、输出控制对象、作用域对象以及流和异常对象等。

本书还介绍了 JSP 访问数据库的内容，包括 JDBC 简介、SQL 包和连接数据库等知识；JSP 中的组件技术——JavaBean 的内容；JSP 中使用 XML 的技术；JSP 的自定义标记库 JSTL 的内容；JSP 的核心技术 Servlet 的内容。

本书最后介绍了一个综合性实例——在线图书商城的制作全过程。内容包括需求分析，数据库设计，设计与实现时对图书类别、图书和评论 3 大模块的信息进行管理等。通过这个综合实例，读者可以在实际开发过程中深入理解 JSP 的内容。

本书特色

本书通过实例介绍 JSP 程序开发知识，具有实用性教程的特色。

- 本书汇总了多年的程序员职业教学培训经验，内容组织合理，实例丰富全面。
- 本书使用 JSP 语言开发了大量实例，读者可以通过这些丰富实例学习 JSP 编程实践知识。
- 本书编写了大量“实验项目”，引导读者应用该章的知识独立练习编程项目。
- 复习题可以帮助学生检查对 JSP 开发理论知识的掌握程度。

读者对象

本书由多家院校的教师在各家院校成熟教案及原有自编教材的基础上整合编写。作者拥有丰富的 JSP 开发案例和教学经验。本书共 11 章，需要 33 个课时。为了给教师授课提供方便，本书提供了教学课件，读者可以从 www.tup.tsinghua.edu.cn 下载使用。

本书可以作为普通高校计算机相关专业 JSP 编程初级教程，也可以作为接触过 JSP 基础知识、需要深入学习 JSP 动态网站开发的中级教程。在编写过程中难免会有漏洞，欢迎读者与我们联系，帮助我们改正提高。

编　者

2006 年 11 月

目 录

CONTENTS

第 1 章　JSP 与 Internet

Java Server Pages（JSP）是基于 Java 语言的服务器端脚本技术。JSP 提供动态网页设计技术，建立在 Java Servlet 技术的基础之上。使用 JSP 技术，程序开发人员可以利用 JSP 的标记在 HTML 静态内容中加入动态数据，制作动态网页。编写 HTML 代码的工作可以由网页设计者负责，而动态数据则由程序开发人员负责，这样可更有利于网页设计者和程序开发人员发挥各自的特长。

本章将从 Web 的发展开始，探讨开发动态 Web 网页的技术，并比较各种技术的优缺点。然后介绍 JSP 的工作原理和 JSP 页面的生命周期，以便加深对 JSP 工作原理的理解。为了运行 JSP 程序，需要安装 JSP 服务器（如 Tomcat），除此之外，为了使创建的 JSP 程序更好地运行，还需要对 JSP 服务器进行配置。最后，介绍超文本传输协议（HTTP），因为 JSP 程序处理的数据主要是通过 HTTP 协议进行传输的。

本章学习要点：

- 了解什么是真正的动态网页以及常用的动态网页技术
- 理解 JSP 的工作原理
- 了解 JSP 页面的各个生命周期
- 编写简单的 JSP 程序，并能够使之在 Tomcat 等 JSP 服务器上运行
- 理解 Tomcat 服务器的配置文件
- 了解 HTTP 协议

1.1 Web 应用演化

在过去的几年间，Web 技术经历了重大演变。原来 Web 还只是由静态文档构成的。现在，不仅可以在 HTML 中嵌入程序，还可以在运行时向 HTML 文档添加动态内容，从而构成动态网页。

1.1.1 静态 Web

静态 Web 是最简单的 Web 结构。图 1-1 显示了静态 Web 框图。在静态 Web 中，客户端使用 TCP/IP 连接到 Web 服务器，并且使用 HTTP 产生请求。服务器接收到该 HTTP 请求后，向客户端发送一个已生成的 HTML 文档。此文档可以是文本、超链接和格式化标签等。它不包含任何动态内容，也不包含用户与之交互的方式。在这种结构中，HTML 文档是 Web 开发人员预先设定好的，它不向客户端提供交互。

随着技术的发展，人们在 HTML 页面中添加样式表、客户端脚本、Flash 动画、Java Applet 小程序和 ActiveX 控件等，使页面的显示效果更加美观、具有动画效果。但是，

这只不过是视觉动态效果，仍然不具备与客户端进行交互的功能。

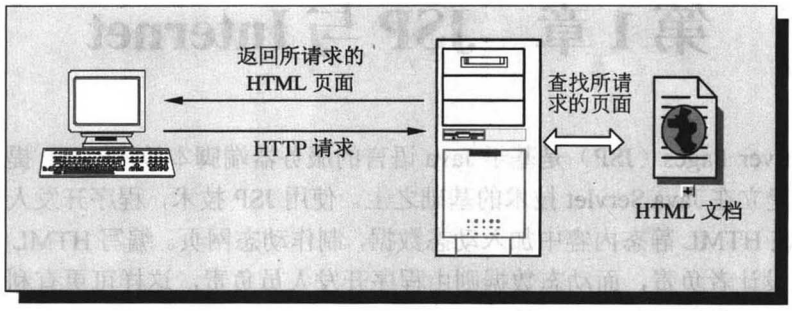


图 1-1 静态 Web 框图

1.1.2 动态 Web

静态 Web 网页上的每一行代码都是由 Web 开发人员预先编写好后，是实实在在存储在 Web 服务器上的，在发送到客户端的浏览器上后不再发生任何变化。动态网页与网页上的各种动画、滚动字幕等视觉上的“动态效果”没有直接关系。这里所说的动态网页可以是纯文字内容，也可以是包含各种动画的内容，这些只是网页具体内容的表现形式，无论网页是否具有动态效果，采用动态网站技术生成的网页都称为动态网页。真正的动态网页体现在“交互性”上，也就是动态网页能根据不同浏览者的请求和访问时间显示不同的内容。

从浏览者的角度来看，无论是动态网页还是静态网页，都是基本的文字和图片信息，但从网站开发、管理、维护的角度来看，两者具有很大的差别。首先，动态网页是在静态网页的基础上，添加服务器端脚本或命令，实现与服务器的交互；其次，动态网页一般以数据库技术为基础，降低网站维护的工作量；最后，采用动态网页技术的网站可以实现更多的功能，如用户注册、登录、在线调查、网上购物、订单管理等。常用的动态网页技术主要有以下几种：CGI、ASP、ASP.NET、PHP、Servlet 和 JSP。

（1）CGI

CGI（Common GateWay Interface，公用网关接口）是用来建立动态网页的技术。当用户在客户端的 Web 浏览器向服务器发送 HTTP 请求时，可以在发送请求的同时附带数据信息。当服务器接收到该请求时，它会调用一个 CGI 程序以响应请求。然后由 CGI 程序负责从请求中提取数据，并对提取的数据进行操作（例如，查询数据库）。最后根据执行结果产生一个 HTTP 响应，此响应一般是根据执行结果生成的新 HTML 文档。图 1-2 演示了这个过程。

CGI 的网页可以用很多语言编写，如 C、C++、VB 和 Perl 语言。它在语言的选择上有很大的灵活性，最常用的 CGI 开发语言为 Perl。它有一个最大的缺点：通常，CGI 对每个 HTTP 请求都产生一个新的进程，当通信量很低时，不会产生问题，但当通信级别增长时，就会造成大量的系统开销，导致系统的运行效率降低。

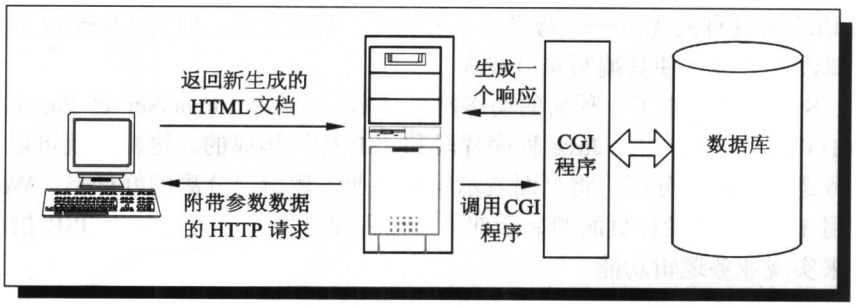


图 1-2 CGI 程序生成动态网页

（2）PHP

PHP（Personal Home Pages）是一种在服务器端运行的、HTML 文档中嵌入的脚本语言。它借鉴 C 语言、Java 语言和 Perl 语言的语法，同时具有自己独特的语法，可以运行于多种平台上。

由于 PHP 采用源代码公开方式，这使得它可以不断地加入新东西，形成庞大的函数库，以实现更多的功能。PHP 几乎支持现在所有的数据库，但是 PHP 没有像 JSP 和 ASP 那样对组件的支持功能，其扩展性较差。

（3）ASP

ASP（Active Server Pages）是微软公司提供的开发动态网页的技术，具有开发简单、功能强大等优点，ASP 使生成 Web 动态内容及构造功能强大的 Web 应用程序的工作变得十分简单。例如，要收集表单中的数据时，只需要将一些简单的指令嵌入到 HTML 文件中，就可以从表单中收集数据并进行分析处理。对于 ASP，还可以便捷地使用 ActiveX 组件来执行复杂的任务，比如连接数据库以检索和存储信息。

ASP 是和平台相关的，即只能运行在 Windows 平台上。除此之外，ASP 程序代码不够结构化，其中混合了 HTML 标记、客户端脚本和服务器端脚本。

（4）ASP.NET

ASP.NET 也是微软公司提供的开发动态网页的技术。ASP.NET 相对于 ASP 是一个革命性的创新。从运行机制上来说，ASP 属于一种解释型的编程框架，它的核心是 VBScript 和 JavaScript 脚本语言，这两种脚本语言决定了 ASP 先天不足，这两种脚本语言无法像传递的编程语言那样进行底层操作，所以有时不得不借助于其他语言编写的组件。ASP.NET 是一种编译型的编程框架，它的核心是.NET Framework。它可以运用 Visual Baxic.NET 和 C#等编程序语言开发，这样就不需要借助于其他组件来完成一些底层操作。在运行速度上，ASP.NET 是先编译后运行，也就是第一次请求时会进行编译，之后的请求就可以在前面的编译结果上直接运行；而 ASP 是解释型脚本语言，每次请求都需要进行解释。ASP.NET 与 ASP 都是微软公司推出的动态网页技术，它与 ASP 一样，最好的运行平台为 Windows 平台。

（5）Servlet 和 JSP

Servlet 和 JSP 都是基于 Java 语言上的一种动态网页技术。Servlet 程序其实就是 Java 程序，只不过它所使用的类库为 Java Servlet API，用于编写服务器端的程序。Servlet 继

承了 Java 语言所具有的优点——跨平台、安全性、易开发等。但是，Servlet 的页面显示和业务逻辑没有分离，使其编写难度较大。

为此，Sun 公司推出了一种新的动态网页接术——JSP（Java Server Pages）。JSP 是在传统的 HTML 页面中加入 Java 程序片段和 JSP 标记构成的。这样，就可以将页面表示层和业务逻辑层进行分离。将页面表示层和为业务逻辑层分离的好处是：Web 开发人员可以使用 HTML 来设计页面的显示部分，而程序开发人员可以使用 JSP 指令和 Java 程序片段来实现业务逻辑功能。

Servlet 和 JSP 作为 J2EE（Java2 企业版）开发平台的一部分，它们最大的好处就是跨平台性，可实现“一次编写，处处运行”。

1.2 JSP 概述

JSP 是由 Sun 公司为创建动态 Web 页面而定义的一种技术。它们是与 Java 相关的一种 HTML 文档，Java 提供动态内容。JSP 是在服务器端应用的一种脚本，它接受请求并生成响应。此请求通常由一个 Web 客户端发送，而响应则是根据请求生成的一种 HTML 文档，该文档需要返回给 Web 客户端。由于 JSP 是服务器端应用，所以它拥有对服务器端资源的访问权限，诸如 Servlet、JavaBean、EJB 和数据库。

1.2.1 JSP 工作原理

JSP 文件是在一个普通的静态 HTML 文件中添加了一些 Java 代码，JSP 文件的扩展名为.jsp。当第一次访问 JSP 页面的时候，这个文件首先会被 JSP 容器翻译为一个 Java 源文件，其实就是一个 Servlet，并进行编译生成相对应的字节码文件.class，然后像其他 Servlet 一样，由 Servlet 容器来处理。Servlet 容器装载这个类，处理来自客户的请求，并把结果返回给客户。这个过程如图 1-3 所示。

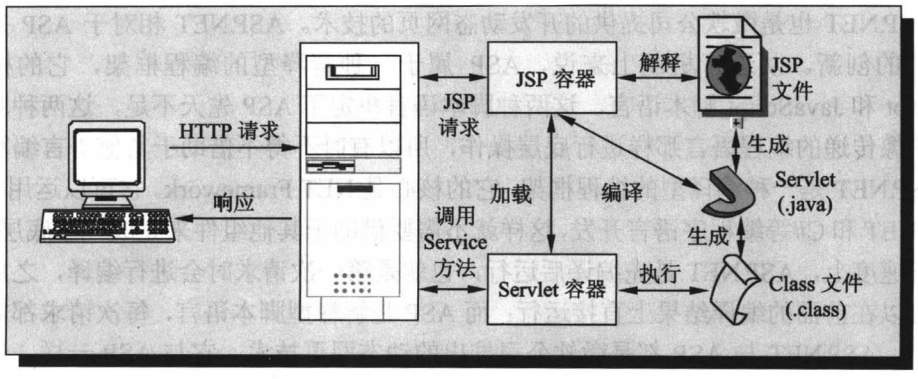


图 1-3 JSP 页面的执行过程

如果以后再有客户访问这个页面时，只要该文件没有发生过更改，JSP 容器就直接调用已经装载的 Servlet。如果已经做过修改，那就会再次执行以上过程，翻译、编译并

装载。因为首次访问的时候要执行一系列上面的过程，所以第一次访问某 JSP 页面时速度会较慢；但在以后运行时速度将非常快。

Java2 企业版（J2EE）定义了几个容器：JSP 容器、Servlet 容器和 EJB 容器（企业级 JavaBean）。J2EE 规范定义的容器用于为企业级组件在其生命周期内或活动期内提供运行环境。J2EE 容器管理组件的生命周期并向组件提供不同的服务。此外，它们还协调组件与更大的运行环境之间的交互。

1.2.2 JSP 页面的生命周期

JSP 页面在被编译为 Servlet 并加载到 Servlet 容器后，Servlet 容器使用 3 个方法控制其生命周期。这 3 个方法为 jspInit()、jspSevice()和 jspDestroy()。这些方法是根据 JSP 页面的状态由 JSP 容器调用的。

在 javax.servlet.jsp 包中定义了一个 JspPage 接口，该接口定义了 jspInit()与 jspDestroy()两个方法。jspInit()与 jspDestroy()方法分别用于完成初始化和释放资源的操作。针对 HTTP 通信协议，javax.sevler.jsp 包定义了一个 HttpJspPage 接口。该接口只定义了一个 jspService()方法，该方法由 JSP 容器调用，以响应客户端的 HTTP 请求。

一般把 jspInit()方法、jspService()以及 jspDestroy() 3 个方法称为 JSP 生命周期方法。当一个 JSP 页面被请求调进时，由 JSP 容器把该 JSP 页面转换成一个 Servlet。在转换成功后，JSP 容器将调用 jspInit()方法，创建 Servlet 的一个实例。

jspInit()方法在 Servlet 生命周期只执行一次，然后将调用 jspService()方法处理来自客户端的请求。当同时有多个客户请求时，JSP 容器将创建该 Servlet 的多个线程响应，这样每个客户请求对应一个 Servlet 线程，以多线程方式执行提高了系统的并发性。由于 Servlet 始终驻留在内存中，所以响应速度非常快。如果在 JSP 页面被转换成 Servlet 后，该 JSP 页面又被修改了，则 JSP 容器会重新编译该 JSP 页面，并用新生成的 Servlet 取代内存中旧的 Servlet。当不再需要一个 Servlet 时，jspDestroy()方法将被调用，以释放该 Servlet 实例占用的系统资源。

1.3 配置 JSP 服务器

自从 JSP 发布以后就出现了各式各样的 JSP 服务器，这里将只介绍一种简单的 JSP 服务器 Tomcat 的安装与配置。由于 Windows 操作系统被广泛使用，且其用户界面较友好，操作方便，因此，选择的 Tomcat 版本为 Windows 版本。Tomcat 可从 http://tomcat.apache.org/ 免费下载。

1.3.1 安装 Tomcat

Tomcat 是一个免费的开源 Servlet 容器，它是 Apache 基金会的 Jakarta 项目中的一个核心项目，由 Apache、Sun 和其他一些公司及个人共同开发而成。由于有了 Sun 的参与和支持，最新的 Servlet 和 JSP 规范总能在 Tomcat 中得到体现。由于 Tomcat 是完全由 Java

编写的，其 Win32 版本与 Linux 版本的安装没有多大的区别。这里以 Win32 版本在 Windows XP 中的安装为例说明其安装过程，具体步骤如下：

（1）运行 apache-tomcat-5.5.17.exe 安装程序，出现如图 1-4 所示的 Tomcat 安装向导窗体。需要注意的是，在安装 Tomcat 之前，首先需要安装 JDK。

（2）单击 Next 按钮，进入 Tomcat 安装协议窗体，如图 1-5 所示。

图 1-4　进入 Tomcat 安装向导

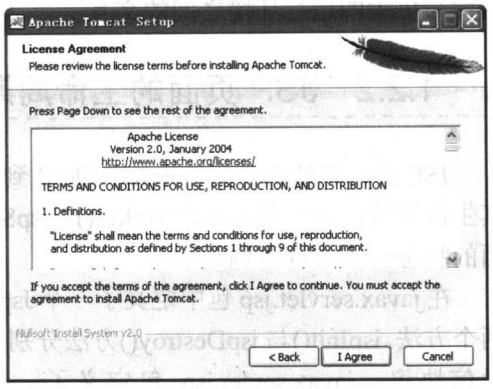


图 1-5　安装协议

（3）单击 I Agree 按钮同意安装协议，进入选择安装方式窗体，如图 1-6 所示。

（4）采用 Tomcat 的默认安装方式，单击 Next 按钮，进入选择 Tomcat 安装路径窗体，如图 1-7 所示。

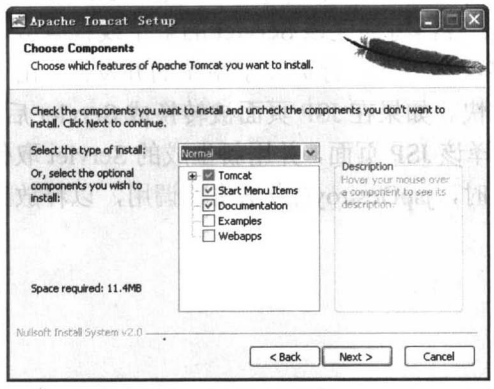


图 1-6　选择安装方式

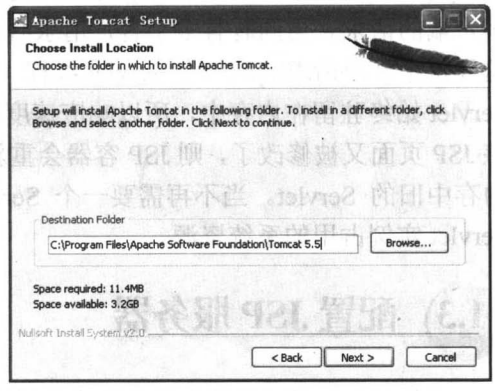


图 1-7　选择安装路径

（5）这里使用其默认的安装路径。单击 Next 按钮进入对 Tomcat 进行基本配置的窗体，如图 1-8 所示。在这里可以设置 Tomcat 使用的端口以及 Web 管理界面的用户名和密码。另一个重要的选择是 HTTP 连接端口号，该端口号是客户连接到 Tomcat 时所使用的端口，应确保该端口未被其他程序占用。

（6）单击 Next 按钮进入选择 Java 虚拟机窗体，如图 1-9 所示。安装程序会自动搜索安装 JDK 时安装的 Java 虚拟路径，如果没有正确显示，则可以手工修改。

（7）单击 Install 按钮，即可安装 Tomcat。成功安装后，程序会提示启动 Tomcat 并查看 readme 文档。Tomcat 正常启动后会在系统托盘区加载图标。用鼠标右击该图标，

可弹出一个快捷菜单。其中，Stop Service 菜单项用于停止 Tomcat 服务，Start Service 菜单项则可以启动 Tomcat 服务。

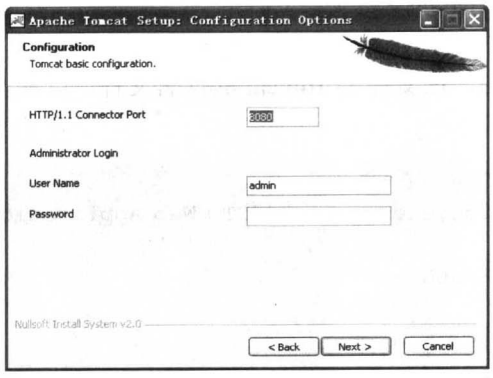


图 1-8 配置 Tomcat

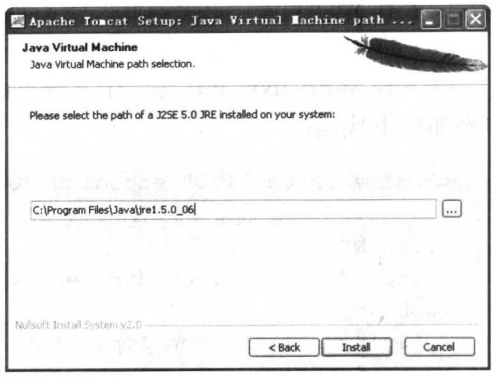


图 1-9 选择 Java 虚拟机的安装路径

（8）至此安装 Tomcat 已经完成，打开浏览器输入“http://localhost:8080”可以看到如图 1-10 所示的 Tomcat 相关信息。

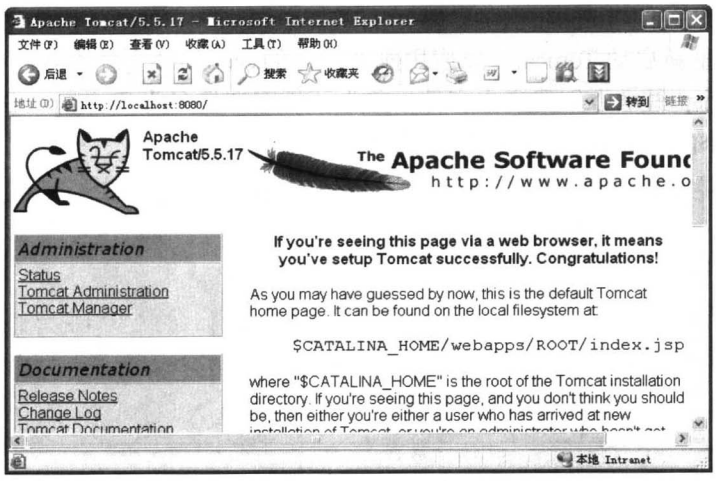


图 1-10 Tomcat 管理页面

1.3.2 第一个 JSP 程序

在安装成功后打开 Tomcat 安装目录，可以看到几个文件夹。其中，Tomcat 将转换后的 Java 文件和 class 文件存放在 work 文件夹下，bin 为 Tomcat 执行脚本目录，conf 文件夹下存放有 Tomcat 的配置文件，lib 文件夹为 Tomcat 运行时需要的库文件，Tomcat 执行时的日志文件存放在 logs 文件夹下，webapps 为 Tomcat 的主要 Web 发布目录。

下面在 Tomcat 目录下创建自己的 Web 应用目录，以便在浏览器中正确显示自己建立的 JSP 程序。

（1）在 Tomcat 安装目录下的 webapps 目录中，可以看到 ROOT、examples、tomcat-docs 之类 Tomcat 自带的目录。

（2）在 webapps 目录下新建一个名称为 myjsp 的文件夹。

（3）在 myjsp 下新建一个文件夹 WEB-INF。注意，目录名称是区分大小写的。

（4）在 WEB-INF 下新建一个文件 web.xml，该文件为 Tomcat 的部署文件，并在其中添加如下代码：

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application
2.3//EN"
"http://java.sun.com/dtd/web-app_2_3.dtd">
<web-app>
<display-name>My Web Application</display-name>
<description>
A JSP application for test
</description>
</web-app>
```

（5）右击系统任务栏右边托盘区中 Tomcat 图标，在弹出的快捷菜单中选择 Stop Service 菜单项，关闭 Tomcat 服务。然后右击 Tomcat 图标，从快捷菜单中选择 Start Service 菜单，重新启动 Tomcat，以应用刚才创建的 web.xml 文件设置。

（6）在 myjsp 目录下创建文本文件，并为其指定文件名为 Test.jsp。注意 JSP 页面的扩展名必须为.jsp。然后在该文本文件中输入如下代码：

```
<%@ page contentType="text/html; charset=gb2312" %>
<html>
<head>
<title>
第一个 JPS 程序
</title>
</head>
<body>
<h2 align="center">
<%=new java.util.Date()%>
</h2>
</body>
</html>
```

下面对这个程序做一个简要说明：所有的 JSP 脚本程序都必须用“<%”和“%>”括起来。为了获取系统的当前日期，使用了 java.util 包中的 Date 类，Data 类可以获取系统的当前时间和日期。

（7）打开浏览器，输入“http://localhost:8080/myjsp/Test.jsp”，这时可以看到如图 1-11 所示的当前系统时间。

图 1-11 显示当前系统时间

为了证明 JSP 的工作原理，可打开 Tomcat

目录下的 work\Catalina\localhost\myjsp\ org\apache\jsp 目录，就会看到一个.java 文件和.class 文件。这就是 Tomcat 服务器解释和编译 JSP 程序后的相关文件。有关.java 文件的内容将在随后讨论的 Servlet 中进行介绍。

1.3.3 配置 Tomcat

Tomcat 提供了一系列的配置文件来帮助用户配置自己的 Tomcat，Tomcat 的配置文件主要是基于 XML 的，如 server.xml、web.xml 等，只有 workers.properties 和 uriworkermap.properties 等几个少数文件是传统的配置文件。本节将详细讨论 Tomcat 的主要配置文件。

1. Tomcat 的主配置文件 server.xml

server.xml 文件描述了 Tomcat 的基本配置信息。server.xml 文件由如表 1-1 中列出的元素组成，各元素及其属性的配置意义如表 1-1 所示。

表 1-1 server.xml 的元素

元素名	属性	说明
Server（该元素是 server.xml 文件最高级别的元素，它描述了一个 Tomcat 服务器）	port	port 指定一个端口，这个端口负责监听关闭 Tomcat 的请求
	shutdown	shutdown 指定向端口发送的命令字符串
Service	name	指定 Service 的名字
Connector（表示客户端和 Service 之间的连接）	port	指定服务器端要创建的端口号，并在这个端口监听来自客户端的请求
	minThreads	服务器启动时创建的用于处理请求的最小线程数
	maxThreads	最多可以创建的处理请求的线程数
	enableLookups	如果为 true，则可以通过调用 request.getRemoteHost() 进行 DNS 查询来得到远程客户端的实际主机名，若为 false 则不进行 DNS 查询，而是返回其 IP 地址
	redirectPort	指定服务器正在处理 HTTP 请求时，如果收到了一个 SSL 传输请求后，重定向的端口号
	acceptCount	指定当所有可以使用的处理请求的线程数都被使用后，可以放到等待处理队列中的请求数，超过这个数的请求将不予处理
	connectionTimeout	以毫秒为单位，指定连接时的超时时间
Engine（表示指定 Service 中的请求处理机，接收和处理来自 Connector 的请求）	defaultHost	指定默认的处理请求的主机名，它至少应该与其中的一个 host 元素的 name 属性值是相同的
Context（每一个 Context 都描述了一个 Tomcat 的 Web 应用程序的目录）	docBase	Context 的目录
	path	表示 Context 在 Web 服务器时的虚拟目录位置和目录名
	reloadable	如果属性为 true，则 Tomcat 会自动检测应用程序的 /WEB-INF/lib 和/WEB-INF/classes 目录的变化，自动装载新的应用程序，可以在不重启 Tomcat 的情况下改变应用程序