

全国高等院校计算机基础教育研究会

“2016年度计算机基础教学改革课题”立项项目

Java Web应用 开发教程 (第2版)

主编◎于静 副主编◎张虹 顾鸿虹



北京邮电大学出版社
www.buptpress.com

全国高等院校计算机基础教育研究会

“2016年度计算机基础教学改革课题”立项项目

Java Web应用 开发教程 (第2版)

主编◎于静 副主编◎张虹 顾鸿虹



北京邮电大学出版社
www.buptpress.com

内 容 简 介

传统 Java Web 应用开发教程的每个实验是相对独立的,学生很难将各个知识点连贯起来,因此很难提高项目实践能力。本书面向应用型本科计算机相关专业,将 Java Web 理论知识与实践环节有机结合,在每一章理论知识后紧跟上机实验环节,全部实验完成一个完整的项目应用。本书系统介绍 Java Web 应用开发过程中的实用技术和标准规范,从 Java 语言基础出发,向读者介绍系统设计与编程思想,让读者能够在一个实际项目中全面而系统地掌握动态网站开发技巧。

本书内容主要包括 Java Web 概述、Java 程序设计基础、网页设计基础、JSP 页面元素、JSP 内置对象、JavaBean 在 JSP 中的应用、EL 表达式语言与 JSTL 标签库、基于 Servlet 的 Web 开发和 Web 设计模式。全书尤其强调实际应用,提高读者运用 Java Web 技术解决实际问题的能力。

本书主要面向 Java Web 初学者,需要读者有一定的 Java 基础。

图书在版编目(CIP)数据

Java Web 应用开发教程 / 于静主编. -- 2 版. -- 北京: 北京邮电大学出版社, 2017. 11

ISBN 978-7-5635-4161-4

I. ①J… II. ①于… III. ①JAVA 语言—程序设计—教材 IV. ①TP312.8

中国版本图书馆 CIP 数据核字 (2017) 第 225212 号

书 名: Java Web 应用开发教程 (第 2 版)

著作责任者: 于 静 主编

责 任 编 辑: 刘春棠

出 版 发 行: 北京邮电大学出版社

社 址: 北京市海淀区西土城路 10 号 (邮编: 100876)

发 行 部: 电话: 010-62282185 传真: 010-62283578

E-mail: publish@bupt.edu.cn

经 销: 各地新华书店

印 刷:

开 本: 787 mm×1 092 mm 1/16

印 张: 18.5

字 数: 458 千字

印 数: 1—1 000 册

版 次: 2010 年 4 月第 1 版 2017 年 11 月第 2 版 2017 年 11 月第 1 次印刷

ISBN 978-7-5635-4161-4

定 价: 41.00 元

• 如有印装质量问题, 请与北京邮电大学出版社发行部联系 •

引 言

对于一名 Java Web 应用开发的初学者来说,主要欠缺两方面能力,一方面是科学良好的系统设计思想,另一方面是实践应用开发能力。本书以此为目的,以 Java 语言为基础,通过实际案例讲解 Java Web 应用开发领域中实际使用的相关技术,并最终帮助读者掌握基于 MVC 设计模式的分层系统设计思想。

本书的读者对象主要包括普通高等院校应用型本科计算机相关专业的学生,综合学生的特点和当前软件行业的就业需求,编者总结实际教学经验与体会,设计了各章节内容及教学重点。书中实例蕴含的思想大部分来自实际的工程项目经验,具有一定的实用性和参考性。

本书在第 1 版的基础上,结合更新了 JDK、Tomcat、MyEclipse 等环境的开发版本,体现 JDK8 之后的新变化,并围绕一个完整的实际项目“网上论坛 BBS 系统”,合理安排了 21 个实验。本书章节结构安排如下。

第 1 章介绍 Web 应用的演化,介绍 JSP 的工作原理,说明搭建 Web 服务器和开发环境的步骤,介绍 MyEclipse 开发工具的使用,并对 HTTP 的特点做简要介绍,编写、部署与运行第一个简单的 JSP 程序。

第 2 章围绕附录所描述的项目需求,讲解 Java 基础的必要知识,包括类和对象的概念,类的封装、继承及多态的特性,抽象类与接口,集合类的使用,使用 JDBC 技术编写数据库连接类,实现 BBS 会员注册和编辑个人信息操作,实现查询版块信息、主题列表信息功能。

第 3 章学习网页 HTML 基础,包括 HTML 文档的基本结构、表格及表单的使用,重点讲解传统表格与 DIV+CSS 两种布局方法,并简要介绍了 JavaScript 脚本语言的客户端验证作用。

第 4 章介绍 JSP 页面元素,包括静态文本、注释元素、指令元素、脚本元素及动作元素,重点讲解实际中使用较多的 page 指令及其表达式、脚本代码的用法。实现 BBS 首页动态显示版块信息,帖子列表页动态显示主题列表,帖子内容页动态显示帖子内容。

第 5 章讨论 JSP 中 9 种内置对象,重点介绍输入输出对象:request、response 和 out 对象,作用域通信对象:session、application 和 pageContext 对象,以及其他对象的用法,并简要介绍 Cookie 的使用。实现 BBS 发表新帖和回复、会员注册并跳转、会员简单登录,实现分页动态显示主题列表和帖子内容、导航栏动态显示、用户登录和登出,实现登录后发表新帖和回复。

第 6 章讲解 JavaBean 组件技术在 JSP 中的应用,介绍 JavaBean 类的典型代码结构, JSP 中使用 JavaBean 的动作元素,将 JavaBean 分为封装数据和封装业务逻辑两种,并以实际案例学习如何使用 JavaBean。

第 7 章学习 JSP 中的 EL 表达式语言与 JSTL 标准标签库的使用,重点讲解 Core 核心标签库和 Format 标签库,并简要介绍自定义标签的创建方法。

第 8 章讨论 Java Web 的核心技术 Servlet,介绍如何创建 Servlet 以及 Servlet 的生命周期,讲解如何在 Servlet 中处理客户端请求并予以响应,如何会话管理进行状态跟踪,介绍以 Servlet 原理为基础的过滤器技术,使用 Servlet 修改 BBS 的会员登录、登出和注册,用户发帖和回帖功能,使用过滤器进行中文处理和身份验证。

第 9 章在前面相关技术的基础上,讲解 Java Web 应用开发的两种模型:模型一(JSP+JavaBean)和模型二(JSP+Servlet+JavaBean),并针对同一案例进行对比分析,总结出基于 MVC 模式的模型二的优势所在。

本书主要面向 Java Web 初学者,需要读者有一定的 Java 基础。

本书第 1 章和第 7 章由顾鸿虹编写,第 2 章、第 8 章和附录由于静编写,第 3 章由杨娜编写,第 4~6 章由张虹编写,第 9 章由冯海燕编写。全书由于静负责内容结构设计和统稿工作。

对于书中的疏漏和不妥之处,恳望读者批评指正。

编 者
2017 年 6 月

目 录

第 1 章 Java Web 概述	1
1.1 Web 应用演化	1
1.2 Web 服务器脚本技术	2
1.3 JSP 工作原理	3
1.3.1 Servlet 技术	3
1.3.2 JSP 生命周期	5
1.4 超文本传输协议	5
1.4.1 HTTP 请求	5
1.4.2 HTTP 响应	6
1.5 本章小结	7
实验 1 安装与配置 JDK、Tomcat	7
实验 2 部署运行第一个 JSP 程序	12
实验 3 配置与使用 MyEclipse	14
第 2 章 Java 程序设计基础	26
2.1 类和对象	26
2.1.1 类的成员	26
2.1.2 对象初始化与构造方法	27
2.1.3 this 关键字	28
2.1.4 包	29
2.2 类的继承	29
2.2.1 父类与子类	30
2.2.2 方法重写	31
2.2.3 super 关键字	32
2.2.4 访问修饰符	33
2.3 抽象类和接口	34
2.3.1 抽象类与抽象方法	35
2.3.2 定义与实现接口	36
2.4 JavaBean 技术	37
2.4.1 封装数据的 JavaBean	37
2.4.2 封装业务的 JavaBean	39

2.5 使用集合类存储对象	40
2.5.1 List 集合	40
2.5.2 Set 集合	41
2.5.3 Map 集合	42
2.6 JDBC 技术	43
2.6.1 java.sql 包	43
2.6.2 创建数据库连接	44
2.6.3 关闭数据库连接	45
2.6.4 Statement 类和 PreparedStatement 类	45
2.6.5 ResultSet 结果集	48
2.7 本章小结	49
实验 4 面向对象的实体类设计	49
实验 5 面向对象的数据访问接口设计	60
实验 6 编写数据库连接类	69
实验 7 实现数据更新操作	74
实验 8 实现数据查询操作	76
第 3 章 网页设计基础	81
3.1 HTML 基础	81
3.2 头部内容	82
3.2.1 <title>标记	82
3.2.2 <base>标记	83
3.2.3 <meta>标记	83
3.3 主体内容	83
3.3.1 文字段落控制	83
3.3.2 图像标记	86
3.3.3 超链接标记	86
3.3.4 表格	87
3.3.5 表单	89
3.4 页面布局	92
3.4.1 表格布局	92
3.4.2 DIV+CSS 布局	93
3.5 JavaScript 的简单应用	95
3.5.1 什么是 JavaScript	95
3.5.2 JavaScript 的事件处理	95
3.5.3 JavaScript 的数据类型和变量	96
3.5.4 JavaScript 的对象及其属性和方法	96
3.5.5 表单验证示例	97
3.6 本章小结	99

实验 9 静态页面布局	99
实验 10 制作表单	106
实验 11 利用 JavaScript 代码进行客户端简单验证	110
第 4 章 JSP 页面元素	112
4.1 JSP 页面基本结构	112
4.2 注释元素	114
4.3 指令元素	115
4.3.1 page 指令	115
4.3.2 include 指令	116
4.3.3 taglib 指令	118
4.4 脚本元素	119
4.4.1 声明	119
4.4.2 表达式	120
4.4.3 脚本代码	121
4.5 动作元素	123
4.5.1 <jsp:include>动作	123
4.5.2 <jsp:forward>动作	124
4.5.3 <jsp:param>动作	125
4.6 本章小结	127
实验 12 使用 JSP 指令	127
实验 13 使用 JSP 脚本元素	129
第 5 章 JSP 内置对象	140
5.1 内置对象介绍	140
5.2 输入输出对象	143
5.2.1 request 对象	143
5.2.2 response 对象	150
5.2.3 out 对象	153
5.3 作用域通信对象	155
5.3.1 session 对象	155
5.3.2 application 对象	160
5.3.3 pageContext 对象	161
5.4 其他对象	164
5.4.1 page 对象	164
5.4.2 config 对象	165
5.4.3 exception 对象	166
5.5 Cookie 的使用	167
5.6 本章小结	169

实验 14 使用 request 对象和 response 对象	169
实验 15 使用 session 对象	173
第 6 章 JavaBean 在 JSP 中的应用	184
6.1 JSP 脚本元素调用 JavaBean	184
6.2 JSP 动作元素与 JavaBean 生命周期	186
6.2.1 <jsp:useBean>动作	187
6.2.2 <jsp:setProperty>动作	187
6.2.3 <jsp:getProperty>动作	188
6.3 封装数据的 JavaBean 与表单交互	189
6.3.1 使用 JavaBean 的表单交互	189
6.3.2 使用 JavaBean 的数据传参	190
6.4 封装业务的 JavaBean 组件	192
6.5 JavaBean 的其他应用	194
6.5.1 基于 JavaMail 的邮件发送 JavaBean	194
6.5.2 使用 JavaBean 实现数据分页显示	197
6.5.3 基于 JSPSmartUpload 的文件上传 JavaBean	202
6.5.4 基于 JGraph 的验证码 JavaBean	203
6.6 本章小结	206
实验 16 使用 JavaBean 封装业务逻辑	206
实验 17 JavaBean 在 JSP 中的使用	209
第 7 章 EL 表达式语言与 JSTL 标签库	211
7.1 EL 表达式语言	211
7.1.1 EL 语法	211
7.1.2 EL 内置对象	213
7.2 JSTL 标签库	215
7.2.1 JSTL 简介	215
7.2.2 添加 JSTL 支持	216
7.3 Core 标签库	217
7.3.1 通用标签	218
7.3.2 条件标签	219
7.3.3 迭代标签	220
7.3.4 URL 相关标签	221
7.4 Format 标签库	223
7.5 SQL 标签库	227
7.6 XML 标签库	228
7.7 自定义标签库	229
7.7.1 自定义标签分类	230

7.7.2 创建自定义标签库	230
7.8 本章小结	236
实验 18 EL 表达式的应用	236
实验 19 JSTL 标记在 JSP 中的使用	238
第 8 章 基于 Servlet 的 Web 开发	241
8.1 Servlet 概述	241
8.1.1 JSP 与 Servlet	242
8.1.2 第一个 Servlet	242
8.1.3 Servlet 生命周期	244
8.2 处理客户请求与响应	246
8.2.1 处理客户表单数据	246
8.2.2 读取 HTTP 请求头信息	249
8.2.3 处理 HTTP 响应头信息	251
8.2.4 Servlet 通信	253
8.3 会话管理	256
8.3.1 会话状态概述	256
8.3.2 会话状态跟踪 API	257
8.4 过滤器	259
8.4.1 创建过滤器	260
8.4.2 解决请求数据中文乱码问题	263
8.5 本章小结	265
实验 20 使用 Servlet 响应客户端请求	265
实验 21 使用 filter 过滤器	268
第 9 章 Web 设计模式	271
9.1 Java Web 应用开发的两种模型	271
9.1.1 模型一:JSP+JavaBean	271
9.1.2 模型二:JSP+Servlet+JavaBean	272
9.2 两种模型案例对比分析	272
9.2.1 问题描述与数据库设计	272
9.2.2 使用模型一实现	273
9.2.3 使用模型二实现	275
9.3 MVC 模式的优点	278
9.4 本章小结	278
附录 A 项目案例分析——网上论坛 BBS 系统	279
参考文献	284

第 1 章 Java Web 概述

【本章要点】

- JSP 工作原理。
- JSP 生命周期。
- Web 服务器和开发环境的搭建。
- 第一个 JSP 程序。
- HTTP 的特点。

随着互联网技术与应用的不断发展,普通的静态网页已不能满足网上信息交流的需求,具有交互功能的动态网页得到了广泛的应用。Web 程序设计技术就是用于实现动态交互式功能的网页制作技术,通过 Web 程序语言(CGI、PHP、ASP、JSP、ASP. Net 等)设计的动态网页可以根据用户的即时操作和即时请求,使网页内容发生相应的变化,从而可以实现功能强大的交互式操作。

本章简要介绍 Web 应用的演变,以及 Servlet 和 JSP 的关系,详细讲解 JDK 的安装及部署过程,以及如何编写及运行第一个 HelloWorld 小程序,最后介绍客户端和服务端通信的 HTTP 的请求和响应。

1.1 Web 应用演化

Web 应用经历了从 C/S 架构到 B/S 架构的演变过程,目前主流网络应用主要采用 B/S 架构。

1. C/S 架构

在 Web 应用普及之前,多数网络应用需要在客户端安装程序,一般称为 C/S 架构,即 Client/Server。这种架构的应用不仅需要编写服务器端的脚本程序,也需要编写客户端程序,用户需要通过安装文件在本机安装,此类应用如 Windows Media Player、QQ、Office 等。

2. B/S 架构

对于 C/S 架构的应用来说,将任务合理分配到 Client 端和 Server 端来实现,因此系统的网络通信开销低,应用服务器运行数据负荷较轻,但是客户端系统升级或功能更新代价高,而且效率低,因此适用于中小型应用程序。另外一种应用架构是 B/S 架构,即 Browser/Server,本机只需安装浏览器软件。B/S 架构的应用程序主要业务逻辑运行在服务器上,只有极少的业务逻辑运行在客户端,主要数据存储在服务器上,极少数据缓存在客户端。这样的客户端简单,系统维护与升级的成本和工作量小。

1.2 Web 服务器脚本技术

设计动态网页的技术有很多,并且这方面的技术在不断地发展变化,它们有各自的特点。

1. CGI

CGI(Common Gateway Interface)是最古老的 Web 编程技术,是一段在服务器端运行的程序,是面向客户端 HTML 页面的接口,就像一座桥,把网页和 Web 服务器中的执行程序连接起来,把 HTML 接收的指令传递给服务器,把服务器执行的结果返回 HTML 页面。

CGI 可以在任何服务器和操作系统上实现,任何程序语言都可以编写 CGI。CGI 属于底层操作,远不及 ASP、JSP 和 PHP 容易。因为 ASP 等都提供良好的运作环境,底层操作对于它们而言是不必要的。如果把 CGI 比作机器语言,那么 ASP 等就是汇编语言了。CGI 的底层操作可以为大型系统创作之所用。

2. PHP

PHP(Personal Home Page)是一种 HTML 内嵌的语言,是一种在服务器端运行的嵌入 HTML 文档的脚本语言,语言的风格类似于 C 语言,现在被很多网站编程人员广泛运用。PHP 独特的语法混合了 C、Java、Perl 以及 PHP 自创新的语法。它可以比 CGI 或者 Perl 更快速地执行动态网页。

PHP 执行效率比较高,而且 PHP 支持几乎所有流行的数据库以及操作系统。但每种数据库的开发语言都完全不同,这就需要开发人员将同样的数据操作用不同的代码写出多种代码库,增加了程序员的工作量。另外,PHP 是完全免费的,所有代码都是开源的,缺少正规的商业支持,无法实现商品化应用的开发。

3. ASP

ASP(Active Server Pages)是微软推出的用以取代 CGI 的技术。简单来说,ASP 是一套服务器端的脚本运行环境,通过 ASP 可以结合 HTML 网页、ASP 指令和 ActiveX 元素建立动态、交互、高效的 Web 服务器应用程序。

ASP 提供了创建交互网页的简便方法,只要将一些简单的指令嵌入 HTML 文件中,就可以从表单中收集数据。ASP 还可以利用 ADO(Active Data Object,微软开发的一种数据访问模型)方便地访问数据库,使开发基于 WWW 的应用系统成为可能。

4. JSP

JSP(Java Server Pages)也是当前比较热门的 Web 技术,是由 SUN 公司发布的。JSP 为创建高度动态的 Web 应用提供了一个独特的开发环境。

JSP 与 Microsoft 的 ASP 技术非常相似。两者都提供了在 HTML 代码中混合某种程序代码、由语言引擎解释执行程序代码的能力。在 ASP 或 JSP 环境下,HTML 代码主要负责描述信息的显示样式,而程序代码则用来描述处理逻辑。普通的 HTML 页面只依赖于 Web 服务器,而 ASP 和 JSP 页面需要附加的语言引擎分析和执行程序代码。程序代码的执行结果被重新嵌入 HTML 代码中,然后一起发送给浏览器。ASP 和 JSP 都是面向 Web 服务器的技术,客户端浏览器不需要任何附加的软件支持。

JSP 的特点如下。

(1) 将内容的生成和显示进行分离。Web 页面开发人员可以使用 HTML 或者 XML 标志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。生成的内容逻辑上被封装在标识和 JavaBean 组件中,并且捆绑在小脚本中,所有的脚本在服务器端运行。如果核心逻辑被封装在标志和 JavaBean 中,那么其他人,如 Web 管理人员或页面设计者,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,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,而又保证了任何基于 HTML 的 Web 浏览器的完全可用性。

(2) 生成可重用的组件。绝大多数 JSP 页面依赖于可重用的、跨平台的组件(JavaBean 或者 Enterprise JavaBean 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。

(3) 采用标识简化页面开发。Web 页面开发人员不都是熟悉脚本语言的编程人员。JSP 技术封装了许多功能,这些功能是在易用的、与 JSP 相关的 XML 标识中,进行动态内容生成所需要的、标准的 JSP 标识能够访问的 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如同标识一样执行特定功能的构件来工作。

(4) 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。

(5) 可靠且移植方便。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,可以使用自己所选择的服务器和工具,更改工具或服务器并不影响当前的应用。

1.3 JSP 工作原理

JSP 页面在运行的时候,首先会将整个 JSP 页面编译成一个 Java 文件,而服务器运行时,实际上是在运行 Java 的类文件,这样就达到了一次编写多处运行的目的。这个类文件就是 Java Servlet。

1.3.1 Servlet 技术

1. Servlet 简介

Servlet 是使用 Java Servlet API 编写的、适合于 B/S 模式的、运行在 Web 服务器端的 Java 类,具有独立于平台和协议的特性,可以生成动态的 Web 页面。Servlet 和客户端的通信采用“请求/响应”模式,其工作原理如图 1.1 所示。

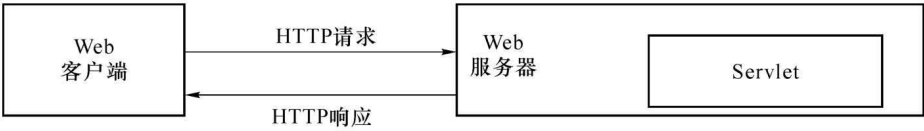


图 1.1 Servlet 的工作原理

2. Servlet 的优点

- (1) 可移植性:Servlet 利用 Java 语言开发,具有 Java 的跨平台性,Servlet 程序可以在任何操作系统上运行。
- (2) 功能强大:包括网络和 URL 访问、通过 JDBC 访问远程数据库、通过对象序列化使用 JavaBean、通过 JNDI 使用 EJB、通过 JXPA 访问 Web 服务等。
- (3) 性能优良:Servlet 程序在加载执行之后,它的实例在一段时间内会一直驻留在服务器的内存中,若有请求,服务器会直接调用 Servlet 实例来服务。并且当多个客户请求一个 Servlet 时,服务器会为每个请求者启动一个线程来处理,所以效率高。
- (4) 可靠性:Servlet 有强类型检查功能,并且利用 Java 的垃圾回收机制避免内存管理上的问题。另外,Servlet 能够安全地处理各种错误,不会因为发生程序上的逻辑错误而导致整体服务器系统的崩溃。

3. JSP 与 Servlet 的关系

JSP 与 Servlet 的关系如图 1.2 所示,可以概括为两点。

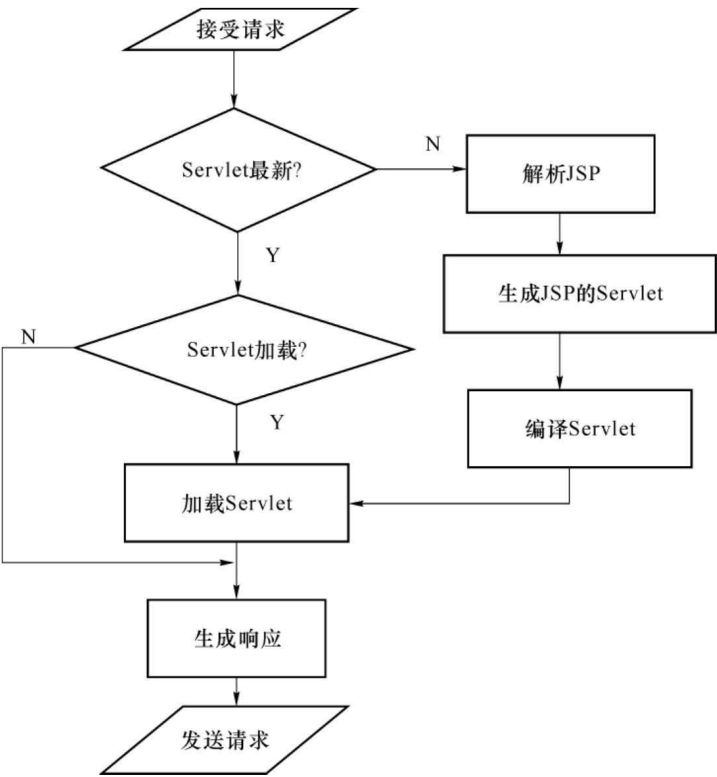


图 1.2 JSP 与 Servlet 的关系示意图

- (1) JSP 的实现是基于 Servlet 的, JSP 页面在运行之前要被解释成 Java Servlet。
- (2) 当 JSP 容器接到对一个 JSP 页面的请求后, 首先判断与 JSP 文件对应的 Servlet 类的名字, 如果该类不存在或比 JSP 文件旧, 容器就会重新创建一个等价的 Servlet 类并编译它。

1.3.2 JSP 生命周期

如果客户端请求的是一个 JSP 页面, 这时服务器会将该页面编译为一个 Servlet 类, 并自动加载形成 Servlet 实例, 把执行结果返回客户端。因此, JSP 网页在执行时会经历以下几个阶段。

(1) 编译阶段

在客户端发出 request 请求后, JSP 容器将 JSP 转译成 Servlet 的源代码, 再将产生的 Servlet 的源代码编译成为 Servlet 类。

(2) 初始化阶段

Servlet 容器加载 Servlet 类, 创建 Servlet 实例并调用 Servlet 的 init 方法进行初始化。

(3) 响应客户请求阶段

对于到达 Servlet 容器的客户请求, 利用容器所创建的 Servlet 实例的对象调用 service 方法处理请求信息, 响应至客户端。

(4) 终止阶段

当 Web 应用被终止, 或 Servlet 容器终止运行, 或 Servlet 容器重新装载 Servlet 的新实例时, Servlet 容器会先调用 Servlet 的 destroy 方法。在 destroy 方法中, 可以释放 Servlet 所占用的资源。

1.4 超文本传输协议

超文本传输协议(HyperText Transfer Protocol, HTTP)是万维网(World Wide Web, WWW, 也简称为 Web)上应用最为广泛的一种网络协议, 是一个属于应用层的面向对象的协议, 适用于分布式超媒体信息系统。HTTP 主要用在客户端(浏览器)和 Web 服务器之间进行通信。所有的 WWW 文件都必须遵守这个标准。

HTTP 于 1990 年提出, 经过十几年的使用与发展, 得到了极大的扩展和完善。HTTP 有 3 个版本, 依次是 0.9、1.0、1.1。目前在 WWW 中普遍使用的版本是 1.1。HTTPng 是发展中的下一代协议, 在效率和性能上会有更进一步的提高。

客户端(浏览器)和 Web 服务器之间要进行通信, 首先使用可靠的 TCP 连接(默认端口为 80), 然后浏览器要先向服务器发送请求信息, 服务器在接收到请求信息后, 作出响应, 返回相应的信息, 浏览器接收到来自服务器的响应信息后, 对这些数据进行解释执行。所以 HTTP 可以分成两部分: HTTP 请求和 HTTP 响应。

1.4.1 HTTP 请求

HTTP 请求格式如下:

```
< request - line>  
< headers>
```

```
< blank line>
[< request - body>]
```

在 HTTP 请求中,第一行必须是一个请求行(request line),用来说明请求类型、要访问的资源以及使用的 HTTP 版本。紧接着是一个头部(header)小节,用来说明服务器要使用的附加信息,如声明浏览器所用语言,请求正文的长度等。在头部之后是一个空行,指示头部结束。在此之后可以添加任意的其他数据,称为主体(body),其中可以包含客户提交的查询字符串信息。

下面是一个 HTTP 请求的例子:

```
GET /sample.jsp HTTP/1.1
Accept:image/gif, image/jpeg, */*
Accept-Language: zh-cn
Connection: Keep-Alive
Host: localhost
User-Agent: Mozilla/4.0(compatible; MSIE 5.01; Windows NT 5.0)
Accept-Encoding: gzip, deflate

username = users&password = 1234
```

在 HTTP 中,定义了大量的请求类型,其中 GET 请求和 POST 请求是最主要的。只要在 Web 浏览器上输入一个 URL,浏览器就将基于该 URL 向服务器发送一个 GET 请求,以告诉服务器获取并返回资源(也就是对网页的访问)。而 POST 请求在请求主体中为服务器提供了一些附加的信息。通常,当填写一个在线表单并提交它时,这些填入的数据将以 POST 请求的方式发送给服务器。

1.4.2 HTTP 响应

HTTP 服务器接到请求后,经过处理,会给予相应的响应信息,其格式与 HTTP 请求相似,如下所示:

```
< status - line>
< headers>
< blank line>
[< response - body>]
```

在 HTTP 响应中与请求唯一真正的区别在于第一行中用状态信息代替了请求信息,状态行(status line)通过提供一个状态码来说明所请求的资源情况。在响应信息的头部也包含很多信息,如服务器类型、日期时间、内容类型和长度等。在头部之后同样需要用一个空行指示头部结束。再接下来的就是服务器返回的内容,如一个 HTML 页面。

下面是一个 HTTP 响应的例子:

```
HTTP/1.1 200 OK
Server: ApacheTomcat/6.0.14
Date: Fri, 20 Nov 2009 10:30:15 GMT
Content-Type: text/html
Last-Modified: Fri, 20 Nov 2009 10:40:25 GMT
Content-Length: 112

< html >
```

```
< head>
< title> HTTP 响应示例</title>
</head>
< body>
Hello HTTP!
</body>
</html>
```

浏览器的每次请求都要求建立一次单独的连接,在处理完每一次的请求后,就自动释放连接。

HTTP 的主要特点可概括如下。

(1) 支持客户/服务器模式。

(2) 简单快速:客户向服务器请求服务时,只需传送请求方法和路径。请求方法常用的有 GET、HEAD、POST。每种方法规定了客户与服务器联系的类型不同。HTTP 简单,使得 HTTP 服务器的程序规模小,因而通信速度很快。

(3) 灵活:HTTP 允许传输任意类型的数据对象。正在传输的类型由 Content-Type 加以标记。

(4) 无连接:无连接的含义是限制每次连接只处理一个请求。服务器处理完客户的请求,并收到客户的应答后,即断开连接。采用这种方式可以节省传输时间。

(5) 无状态:HTTP 是无状态协议。无状态是指协议对于事务处理没有记忆能力。缺少状态意味着如果后续处理需要前面的信息,则它必须重传,这样可能导致每次连接传送的数据量增大。另外,在服务器不需要先前信息时它的应答就较快。

1.5 本章小结

本章介绍了 Web 应用由静态到动态的演化过程、Web 服务器端常用的脚本技术。JSP 和 Servlet 技术是 Java Web 应用开发最重要的两项技术,本章重点剖析了 JSP 工作原理,以及 JSP 与 Servlet 之间的关系。HTTP 是应用最为广泛的一种网络协议,本章简单介绍了该协议诸如 GET、POST 常用请求方法,以及 HTTP 无状态的特点。

实验 1 安装与配置 JDK、Tomcat

一、实验内容

- 练习 JDK、Tomcat 的安装与环境变量配置。
- 在 JDK 环境下编写执行简单的 Java 程序。
- 运行与测试 Tomcat 服务器。

二、实验步骤

1. 安装 JDK

JDK(Java Development Kit)软件开发工具包是整个 Java 的核心,包括 Java 运行环境