



“十二五”职业教育国家规划教材
经全国职业教育教材审定委员会审定

高等应用型人才培养规划教材

基于工作任务的Java Web 应用教程(第2版)

覃国蓉 主 编

周德伟 毛树生 叶建锋 黄晓伟 廖先锋 副主编



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

“十二五”职业教育国家规划教材
经全国职业教育教材审定委员会审定
高等应用型人才培养规划教材

基于工作任务的 Java Web 应用教程（第2版）

覃国蓉 主编
周德伟 毛树生 叶建锋 黄晓伟 廖先锋 副主编

电子工业出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本教材围绕开源的技术示范项目 PetStore 的实现介绍 Java Web 应用系统开发技术: HTML/CSS, JDBC, JSP, 标签库 (JSTL), Servlet, JavaBean, Filter, 以及 J2EE 轻量级框架技术 Hibernate, Struts2 和 Spring, 并且融入面向对象程序设计思想和 MVC 设计模式。本教材从完成最简单的静态版本 (HTML/CSS) 开始, 逐步迭代, 到最后使用框架技术 (Hibernate, Struts2 和 Spring) 完成项目, 难度推进合理。

本教材适合作为应用型本科、高职软件技术及相关专业学生学习 Java Web 应用开发技术的教材, 也可作为面向就业的实习实训教材。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有, 侵权必究。

图书在版编目 (CIP) 数据

基于工作任务的 Java Web 应用教程/覃国蓉主编. —2 版. —北京: 电子工业出版社, 2015.11

“十二五”职业教育国家规划教材 高等应用型人才培规划教材

ISBN 978-7-121-27463-3

I. ①基… II. ①覃… III. ①JAVA 语言—程序设计—高等职业教育—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2015) 第 255731 号

策划编辑: 吕 迈

责任编辑: 吕 迈

印 刷: 涿州市京南印刷厂

装 订: 涿州市京南印刷厂

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1 092 1/16 印张: 13.75 字数: 352 千字

版 次: 2009 年 12 月第 1 版

2015 年 11 月第 2 版

印 次: 2015 年 11 月第 1 次印刷

印 数: 3 000 册 定价: 35.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

/ 前 言 /

项目教学法起源于美国，盛行于德国，尤其适合于职业技术教育。项目教学法成功与否，项目的选择和设计尤为关键。项目通常有 3 种来源：从企业引入的真实项目；教师自己设计的虚拟项目；教材上别人设计的项目。从企业引入的项目直接作为教学项目具有重点不突出、工作量过大和由于工期造成的代码的可读性无法保证的问题；由于很多教师没有相关项目经验，使得教师自己设计的虚拟项目和一些教材上的项目的代码质量、实现的技术和方法与企业真实情况可能有很大的差距。

成功的开源软件由众多优秀程序员共同完成，包含了他们的最佳实践经验，其代码质量、实现的技术和方法要明显优于前面提到的 3 类项目。Java 开源社区产生了许多有价值的开源项目，并且培养了一大批优秀的大师级编程专家，普通的开发者通过这些社区受益多多，就是很好的证明。

PetStore（宠物商店）是 Java 厂商 SUN 公司推出的用于展示 Java EE 技术的示范项目，后来开源社区又推出了它的不同版本以示范各开源技术。该系统的不同版本是世界各地优秀程序员智慧的结晶，其中 JPetStore 设计和架构更优良，各层定义清晰，而 Hibernate JPetStore 增加了 Hibernate 框架技术，所以是当前学习 Java Web 应用开发相关技术的绝好例子。

一线授课教师与企业一线技术人员合作，对 Hibernate JPetStore 进行教学适用化改造，并作为贯穿本教材的案例。教材引导学生在实现该系统的过程中掌握 Java Web 应用系统的开发技术，获得软件开发经验，具备开发实际软件项目的能力，为成为合格的 Java 软件工程师打下基础。

本教材具有以下特色：

（1）选择技术示范项目作为案例，真正实现了“够用为度”。Java Web 开发技术一直在不断发展，Java Web 应用开发技术要讲解到什么程度，无疑其技术示范项目最有发言权。

（2）采用基于原型迭代的软件开发方法的教学法，符合学生和课程的特点。与 PetStore 项目相关的有 10 个任务，从最简单的 HTML+CSS 实现宠物分类展现模块的静态版本开始，每一章中讲解的新技术和方法都是建立在前一章的基础之上的，从而使学生能够循序渐进地学习，到最后能够编写出 Java 高手编写的代码。

（3）“陈述性知识”和“过程性知识”并重。本教材选择开源项目作为贯穿本教材的案例项目，学生不光可以学到相关技术（陈述性知识），还可以学到优秀程序员的经验（过程性知识）：优秀的代码及编码规范、设计技巧和编程模式。

（4）本教材在出版前，已经作为省精品课程和骨干校建设网络课程的配套教材以校本教材的方式使用了 4 次，适用性好。新的版本会跟踪新技术进行内容调整，并且每次使用后，作者都会根据使用效果排除教材的错误，对各章的内容展开方式和描述方式等

进行便于自学和教学的调整，使学生方便自学，教师便于组织教学。

本教材试用版为广东省精品资源共享课程和国家骨干校建设网络课程“轻量级 J2EE 应用开发”的配套教材，经过多轮试用并进行了新技术更新、教学适用性的调整后才正式出版，而且相关课程资源持续更新，学习和教学时可以共享。

为教师授课提供方便，本书提供了多媒体课件、教学案例代码和习题答案，可在电子工业出版社的华信教育资源网免费下载（<http://www.hxedu.com.cn>）。

本教材由覃国蓉主编，周德伟、毛树生（企业）、叶建锋、黄晓伟、廖先锋副主编。其他参编人员有：张璐、任亚洲、陈亚敏、王晨曦、杨海红、杨永滨、徐雪琼、罗贤平、刘红秀、许依达、毛越。

特别感谢：

- Pprun，他是 Hibernate JPetStore 的实现者，本教材的案例项目就是在对 Hibernate JPetStore 改造的基础上完成的，他还在开源技术的使用方面给予了我们很多宝贵经验，正是有了 Pprun 这些开源技术爱好者的无私奉献，Java 技术才得以快速发展和广泛使用。
- 广东省精品课程团队何涛老师、刘志军老师、杨海红老师给予的宝贵建议。
- 深圳信息职业技术学院 2011 软件技术 3-1 班，2012 软件技术 3-1 班，2013 软件技术 3-1 和 3-2 班以及 2014 软件技术 3-1 班对教材的试用和反馈。

编 者

2015 年 8 月

目 录

CONTENTS

第 1 章 背景知识	1	2.2.5 CSS 的盒子模式	21
1.1 Web 应用程序基本概念	1	2.2.6 CSS 的常用属性	21
1.1.1 什么是 Web 应用程序	1	2.3 宠物分类展现的页面及 Web	
1.1.2 静态资源和动态资源	2	应用开发步骤	23
1.1.3 Web 服务器	2	2.3.1 宠物分类展现的页面	23
1.2 理解 HTTP 协议	3	2.3.2 使用 MyEclipse 开发 Web	
1.2.1 HTTP 请求消息格式	3	应用的步骤	24
1.2.2 HTTP 响应消息格式	5	2.4 宠物商城术语表	26
1.3 Java Web 应用开发技术	5	2.5 实现主页面 Main.html	27
1.3.1 静态网页开发技术	5	2.5.1 主页面的左边导航条	
1.3.2 动态网页开发技术	6	部分代码	27
1.3.3 SSH 框架——Java 轻量级企业		2.5.2 主页面的图片导航代码	28
应用解决方案	6	2.5.3 通过层 DIV 标记对主页	
1.4 Servlet	7	面 Main.html 进行布局	30
1.5 JSP	9	2.5.4 通过 CSS 设置效果	30
1.6 PetStore 项目简介	10	2.6 实现品种列表页面主体部	
作业	11	分 Category.html	32
任务 1 开发 1 个简单的个人网站	12	作业	35
第 2 章 使用 HTML 与 CSS	14	任务 2 用 HTML+CSS 实现宠物商城	
2.1 HTML	14	catalog 模块的静态网页版本	35
2.1.1 HTML 文档结构	14	第 3 章 使用 JDBC	37
2.1.2 HTML 标记的公共属性	15	3.1 catalog 模块数据准备	37
2.1.3 HTML 常用标记	15	3.1.1 在 MySQL 中创建一个	
2.1.4 HTML 表单	17	数据库 petstore 及其表	38
2.2 CSS	18	3.1.2 插入测试数据	39
2.2.1 CSS 分类	18	3.1.3 为宠物商城系统创建一个	
2.2.2 CSS 的语法	19	访问数据库 petstore 的用户	41
2.2.3 CSS 的选择器	20	3.2 JDBC 数据库编程	41
2.2.4 CSS 的伪类	20	3.2.1 安装 MySQL 的驱动程序	42
		3.2.2 JDBC 应用程序的模板代码	42

3.2.3 编写 JDBC 应用程序修改数据库.....	44
3.2.4 编写封装创建数据库连接的类.....	46
3.3 POJO+DAO 访问数据库的编程模式.....	47
3.3.1 编写表结构对应的 POJO 类.....	48
3.3.2 设计访问各表的 DAO 类.....	49
3.3.3 编写访问各表的 DAO 类.....	51
3.3.4 DAO 类的使用.....	53
作业.....	54
任务 3 为 catalog 模块准备数据并完成各表对应的 DAO 类.....	55
第 4 章 使用 JSP	58
4.1 JSP 语法元素.....	58
4.1.1 指令标签.....	59
4.1.2 声明标签.....	60
4.1.3 脚本标签.....	60
4.1.4 表达式标签.....	61
4.1.5 动作标签.....	62
4.1.6 注释标签.....	63
4.2 JSP 网页是 Servlet.....	64
4.2.1 JSP 网页是 Servlet.....	64
4.2.2 理解转化单元.....	64
4.3 理解 page 指令标签属性.....	65
4.4 JSP 常用内部对象.....	65
4.4.1 request 与请求参数.....	66
4.4.2 out.....	67
4.4.3 session.....	67
4.5 catalog 模块网页动态版本开发准备.....	67
4.5.1 实现思路.....	67
4.5.2 在 web.xml 中设置欢迎页面.....	68
4.6 用 JSP 实现 Category.jsp.....	69
4.6.1 网页顶部文件 IncludeTop.jsp.....	69
4.6.2 IncludeBottom.jsp.....	72
4.6.3 用 JSP 实现 Category.jsp.....	72
作业.....	74
任务 4 用 JSP+POJO+DAO+DB 实现 catalog 模块的动态网页版本.....	77
第 5 章 使用 JavaBean/ EL/JSTL/Servlet/统一业务接口	79
5.1 JavaBean.....	79
5.1.1 JavaBean 简介.....	79
5.1.2 在 JSP 中使用 JavaBean.....	80
5.1.3 使用 JavaBean 的优势.....	81
5.2 EL 表达式.....	83
5.2.1 EL 表达式简介.....	83
5.2.2 在 EL 表达式中使用隐式对象.....	84
5.2.3 EL 属性和集合访问操作符.....	85
5.2.4 EL 算术运算操作符.....	85
5.2.5 EL 关系和逻辑运算符.....	86
5.3 使用 Java 标准标签库 (JSTL).....	87
5.3.1 JSTL 标签简介.....	87
5.3.2 获得和安装 JSTL.....	87
5.3.3 常用 JSTL 标签.....	88
5.4 优化宠物分类展现页面.....	94
5.4.1 使用 <jsp:useBean> 去掉宠物分类展现页面中的 new 语句.....	94
5.4.2 用 EL 表达式和 JSTL 标签简化宠物分类展现页面代码.....	94
5.4.3 通过迭代使用 EL 表达式点符号简化对象属性的输出.....	96
5.5 JSP Model1、JSP Model2 及 Servlet.....	98
5.5.1 JSP Model1.....	98
5.5.2 Servlet.....	99
5.5.3 使用 Servlet 去掉 PetStore 宠物分类展现页面中的 Java 代码.....	99

5.6 使用统一的业务接口	102	7.3.3 创建 CartServlet 相关属性 和方法实现“添加到购物车” 功能并配置	130
5.6.1 设计一个系统共享的业务 接口 PetStore	102	7.3.4 购物车页面/cart/Cart.jsp 的 实现	132
5.6.2 设计接口 PetStore 的实现 类 PetStoreImpl	103	7.4 “从购物车删除”与“更新购物车” 的实现	134
5.6.3 用 PetStoreImpl 实现宠物分 类展现各页面	104	7.4.1 实现 removeItemFromCart 方法	134
作业	105	7.4.2 实现 updateCartQuantities 方法	134
任务 5 使用 JSTL/Servlet/EL/JavaBean 优化 catalog 的页面代码	108	作业	135
第 6 章 使用过滤器	110	任务 7 完成宠物商城的购物车功能	136
6.1 什么是过滤器	110	第 8 章 使用 Hibernate	137
6.1.1 过滤器工作原理	111	8.1 Hibernate 简介	137
6.1.2 过滤器的使用	111	8.2 使用 Hibernate 的准备工作	138
6.1.3 过滤器的例子	112	8.2.1 用菜单命令安装配置 Hibernate 开发环境	138
6.2 过滤器编程接口	113	8.2.2 用 DB Browser 创建 POJO 类和映射文件	140
6.2.1 javax.servlet.Filter 接口	114	8.3 用 Hibernate 访问数据库	147
6.2.2 javax.servlet.FilterConfig 接口	114	8.3.1 Hibernate 的编程模式	147
6.2.3 javax.servlet.FilterChain 接口	115	8.3.2 使用 Hibernate 实现数据的 插入	148
6.2.4 请求和响应包装类	115	8.3.3 使用 Hibernate 实现数据的 删除和修改	149
6.3 在 web.xml 中配置过滤器链	116	8.3.4 使用 Hibernate 实现数据的 加载	150
6.4 高级特性	118	8.3.5 使用 Hibernate 实现数据的 查询	151
6.4.1 使用响应包装类	118	8.4 使用 Hibernate 重写 DAO 类	152
6.4.2 关于过滤器的重要内容	122	8.4.1 使用 Hibernate 重写 BaseDao 类	152
6.4.3 过滤器充当 Controller 的 优势	122	8.4.2 BaseDao 类的使用	155
作业	123	8.4.3 基于 BaseDao 改写 CategoryDao 类	155
任务 6 使用过滤器解决宠物商城项目 中的中文乱码问题	124	作业	156
第 7 章 实现购物车模块	125	任务 8 用 Hibernate 优化的宠物分类 展现 DAO 类	157
7.1 购物车的页面及流程	125		
7.2 购物车实现思路	126		
7.3 “添加到购物车”功能的实现	127		
7.3.1 定义 CartItem 类	127		
7.3.2 定义 Cart 类	128		

第 9 章 使用 Struts 2 158

9.1 Struts 2 工作原理..... 158

9.1.1 Struts 1 的局限性及

Struts 2..... 158

9.1.2 Struts 2 的工作流程 159

9.2 用 Struts 2 开发 Web 应用程序 159

9.2.1 安装配置 Struts 2 159

9.2.2 编写 Action 类..... 161

9.2.3 配置 Action 类..... 162

9.2.4 编写用户界面 (JSP 页面) · 164

9.3 Struts 2 的其他重要知识点..... 165

9.3.1 Struts 2 的标签库 165

9.3.2 Struts 2 的类型转换 165

9.3.3 Struts 2 的数据验证 166

9.3.4 Struts 2 的拦截器 166

9.3.5 文件的上传和下载..... 166

9.3.6 动态方法调用..... 167

9.3.7 防止表单的重复提交..... 167

9.3.8 Struts 2 中 Action 与 Servlet

容器的耦合 168

作业..... 168

任务 9 使用 Struts 2 优化宠物分类

展现功能..... 169

第 10 章 使用 Struts 2 进阶 170

10.1 用户登录页面和 MVC 模块划分... 170

10.1.1 用户登录的页面及流程..... 170

10.1.2 用户登录的实现思路..... 171

10.2 用户登录 Model 层的实现 172

10.2.1 创建数据库表 account,

生成对应 POJO 类及

Hibernate 映射文件 172

10.2.2 创建表 account 对应数据库

访问类 AccountDao..... 177

10.2.3 在 PetStore 及其实现类中

增加相关方法或成员变量... 177

10.3 用户登录 View 层的实现 178

10.3.1 用户登录页面..... 178

10.3.2 用户登录成功页面..... 179

10.3.3 用户登录失败页面..... 180

10.4 用户登录 Controller 层的实现..... 181

10.5 为用户登录页面增加数据验证 184

10.6 用户登录功能的相关配置 185

10.6.1 在 web.xml 中配置 Struts 2

过滤器..... 185

10.6.2 创建 struts-account.xml

完成登录退出 186

10.6.3 修改 struts-account.xml

完成数据校验 187

10.6.4 修改 struts-account.xml 完成

防止表单重复提交 187

作业..... 188

任务 10 使用 Struts 2 实现登录注册

账户编辑功能 188

第 11 章 使用 Spring 189

11.1 Spring 简介..... 189

11.1.1 Spring 简介 189

11.1.2 Spring 开发环境的安装

配置..... 189

11.1.3 Spring 的控制反转和依赖

注入 192

11.2 使用 Spring 的依赖注入

重写 catalog 模块..... 192

11.2.1 用 Spring 管理 PetStoreImpl

和各 DAO 类对象之间的

依赖..... 192

11.2.2 生成 BaseAction 传递

petstore 对象 195

11.2.3 重写已经完成的 Action 196

11.3 使用 Spring 简化 Hibernate 编程... 196

11.3.1 继承 HibernateDaoSupport

实现 BaseDao 类..... 197

11.3.2 在 Spring 配置文件中

注入 sessionFactory 198

11.3.3 使用 import 简化配置

文件..... 200

11.4 增加分页显示功能	202	11.4.4 修改相关 JSP 页面	206
11.4.1 分页显示的实现思路	202	作业	207
11.4.2 使用 Spring 的 PagedListHolder 进行分页	203	任务 11 用 Spring 改写 Catalog 和用户 登录模块	208
11.4.3 修改相关 Action	204	参考文献及网址	210

本章要点

介绍 Web 应用程序的相关概念：Web 应用程序；静态资源和动态资源；Web 服务器

介绍 HTTP 协议相关知识：HTTP 协议；HTTP 请求消息格式及三种重要的 HTTP 请求方法 GET、POST 和 HEAD；HTTP 响应消息格式

介绍 Java Web 应用开发相关技术：静态网页开发技术；动态网页开发技术；SSH 框架（Java 轻量级企业应用解决方案）

介绍 Servlet 的相关概念：Servlet，Servlet 容器；开发 Servlet 的过程；Servlet API

介绍 JSP 的相关概念：JSP；JSP 的优势

介绍本教材使用案例系统：Petstore 宠物商城

1.1 Web 应用程序基本概念

1.1.1 什么是 Web 应用程序

Web 应用程序是通过 Web 网站提供服务的应用程序。最常见的例子是提供邮件服务的网站，如 163 网易免费邮箱。

最初的 Web 网站，所有网页的内容都是静态的 HTML 网页。在这种情况下，用户只能从 Web 服务器提取静态的页面信息并显示到浏览器中，网站所能实现的任务仅仅是静态的信息显示，而不能与客户产生互动。

JSP 和 Servlet 等技术的出现，使得 Web 服务器的功能得到扩展，可以根据用户的输入信息产生对应的页面（动态网页）。这时的服务器端不只是接收请求并返回页面，还可以处理复杂的业务逻辑并能访问数据库，提供的功能也越来越强大，Web 逐渐成为应用程序开发的首选平台。

1.1.2 静态资源和动态资源

用静态技术（如 HTML）实现的资源为静态资源（或静态页面），静态资源没有处理能力。用动态技术（如 JSP 和 Servlet）实现的资源为动态资源（或动态页面），动态资源具有处理能力。

当浏览器请求 `www.myserver.com/myfile.html` 页面时，Web 服务器找到 `myfile.html` 文件直接返回给浏览器，`myfile.html` 文件是一个静态资源。当浏览器请求 `www.myserver.com/reportServlet` 时，`reportServlet` 是一个 Servlet，是具有处理能力的动态资源，Web 服务器会将该请求传递给 `reportServlet`，该 Servlet 会进行处理（如查询数据库），然后根据处理结果动态生成一个 HTML 文件给 Web 服务器，最后由 Web 服务器将生成的 HTML 文件返回给浏览器。

通常 Web 应用程序既有静态资源，又有动态资源，但是只有动态资源使得 Web 应用程序区别于最初的网站，具有业务处理能力，从而称得上是应用程序。

我们需要区分一下 URI、URL 和 URN。根据 W3C 的定义，Web 上可用的每种资源——HTML 文档、图像、视频片段、程序等，由 URI（uniform resource identifier，通用资源标识符）进行标识，而 URL 是 uniform resource locator，统一资源定位器，除了可以用来标识一个资源，而且还指明了如何访问这个资源。`http://www.manning.com/files/sales/report.html` 就是一个 URL，它唯一标识某资源并规定了如何访问它（即资源的网址）。其他 URI，比如：`mailto: cay@horstman.com`，根据该标识符无法定位任何资源。像这样的 URI 我们称之为 URN（统一资源名称），所以 URI 是一个相对来说更广泛的概念，URL 是 URI 的一种，是 URI 命名机制的一个子集，可以说 URI 是抽象的，而具体要使用 URL 来定位资源。比如后面章节中的 `taglib` 指令标签 `<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>`，其中的 `uri` 也不是一个网址 URL，只是唯一标识一个资源。

1.1.3 Web 服务器

Web 应用程序存放在 Web 服务器上。常用的 Java Web 服务器有 Tomcat（Apache 基金会）、Resin（Caucho Technology 公司）、JRun（Macromedia 公司）、WebLogic（BEA 公司）和 WebSphere（IBM 公司），其中 Tomcat 是由 Apache 基金会开发的一个免费的 Web 服务器。本教材采用 Tomcat 作为 Web 服务器。

每个 Web 应用程序都有一个很重要的配置文件：部署描述文件 `web.xml`，它包含了对 Web 应用程序的所有动态组件的描述，如这个文件对 Web 应用程序的每个 Servlet 都进行

zz 配置。Web 服务器根据 web.xml 对 Web 应用程序的 Web 组件初始化，使得它们可以被客户端访问。

1.2 理解 HTTP 协议

Web 应用程序的客户端和服务端通过 HTTP 协议进行通信。HTTP 协议是基于请求-响应的无状态协议。客户端发送 HTTP 请求到服务器端请求资源，服务器端通过 HTTP 响应返回请求的资源，如图 1.1 所示。客户端打开一个连接并发送一个 HTTP 请求消息；客户端接收服务器端发送的 HTTP 响应并关闭连接，即一旦回答了请求，服务器就关闭连接，在客户端和服务端上没有存储连接信息，所以 HTTP 是无状态协议。

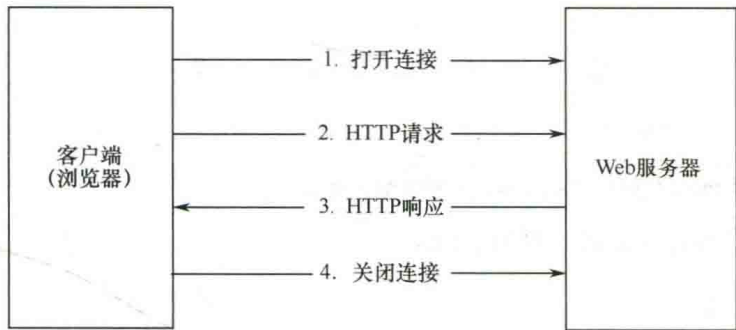


图 1.1 HTTP 协议

在 Internet 中，浏览器就是 HTTP 客户端，Web 服务器是 HTTP 服务器，资源就是 HTML（JSP、ASP 或 PHP）页面、图片文件、Servlet 等。

1.2.1 HTTP 请求消息格式

客户端发送给服务器端的 HTTP 消息叫 HTTP 请求消息，本节介绍 HTTP 请求消息格式。

客户端发送的请求消息为文本流，由以下内容组成：

- 请求行（initial line 或 request line）。该请求消息的第一行称为请求行，包括方法字段、URI 字段、HTTP 版本字段。
- 头部行（Header lines）。请求行后续各行都称为头部行。
- 空白行（Blank line）。表示请求结束。
- 附属体（data）。用于 POST 方法。

```
GET /reports/sales/index.html HTTP/1.1
Host:localhost:8080
User-agent:Mozilla/5.0
```

Accept-language:zh-cn

（额外的回车符和换行符）

上面这个消息是用 GET 方法，表示向服务器请求资源，/reports/sales/index.htm 为请求的资源的本地路径，HTTP/1.1 说明浏览器正在使用的是 HTTP 协议版本。

请求行的方法字段有若干个值可供选择，包括 GET、POST 等，下面分别介绍。

1. GET 方法

HTTP 请求消息绝大多数使用 GET 方法，这是浏览器用来请求资源的默认方法，所请求的资源就在 URL 字段中指定，如果是动态资源，其需要的参数通过在 URL 后增加查询字符串（又叫请求参数）来传递。图 1.2 为传递 userid 参数值 john 的请求行。

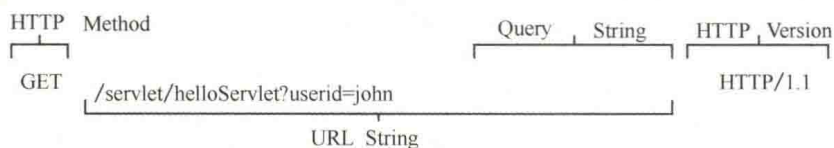


图 1.2 传递 userid 参数值 john 的请求行

请求行问号？后面的字符串为请求字符串，它是由&号隔开的名称值对。如：

name1=value1&name2=value2&...&nameM=valueM

图 1.2 中参数名为 userid，值为 john。

2. HEAD 方法

HEAD 方法与 GET 方法类似，两者的差别只是服务器在对 HEAD 方法的响应消息中去掉了所请求的对象，其他内容则与对 GET 方法的响应消息一样。HEAD 方法通常用于 HTTP 服务器软件开发人员进行调试。

3. POST 方法

POST 方法用于向服务器的动态资源发送待处理的数据，动态资源由请求行 URI 字段指定，而数据则通过附属体来传递。POST 方法适用于需由用户填写表单的场合，用户提交表单后，浏览器就像用户单击了超链接那样仍然从服务器请求一个 Web 页面，不过该页面的具体内容却取决于用户填写在表单各个字段中的值。如果浏览器使用 POST 方法提出该请求，那么请求消息附属体中包含的是用户填写在表单各个字段中的值。图 1.3 为提交表单产生的 POST 请求消息，其中 ContentLength 的值为 data 字段的长度。



图 1.3 提交表单产生的 POST 请求消息

可见 POST 消息参数的传递不在请求行，而在附属体 Data 中，所以在地址栏中看不

到提交的数据信息，并且使用 POST 方法提交数据，没有数据量的限制。虽然 GET 方法是浏览器默认的提交方法，但是我们编写程序的时候，出于对数据的安全性考虑，在没有明确要求用 GET 方法提交数据的时候，尽可能使用 POST 方法，这样做有 2 点好处：一是增加安全性，因为请求数据在地址栏内不可见；二是不用考虑数据容量的问题，因为 POST 请求提交的数据在理论上没有长度的限制，而有些浏览器和服务对请求字符串长度有限制（不能超过 255 个字符）。

1.2.2 HTTP 响应消息格式

服务器端返回给客户端的 HTTP 消息叫 HTTP 响应消息。HTTP 响应消息的第一行（initial line）叫状态行（status line）。状态行有 3 个字段：协议版本字段、状态码字段、原因短语字段。

下面是一个典型的 HTTP 响应消息：

```
HTTP/1.1 200 OK
Connection:close
Date: Thu, 13 Oct 2005 03:17:33 GMT
Server: Apache/2.0.54 (Unix)
Last-Modified: Mon, 22 Jun 1998 09:23:24 GMT
Content-Length: 6821
Content-Type: text/html
```

（数据 数据 数据 数据 数据……）

这个响应消息分为 3 部分：1 个起始的状态行（status line），6 个头部行、1 个包含所请求对象本身的附属体。本例的状态行表明，服务器使用 HTTP/1.1 版本，响应过程完全正常（也就是说服务器找到了所请求的对象，并正在发送）。

1.3 Java Web 应用开发技术

1.3.1 静态网页开发技术

Java Web 应用开发常用的静态网页开发技术包括：

- HTML 语言
- 美化网页的 CSS
- 增加客户端处理能力的 JavaScript
- 提供良好用户体验的 AJAX

HTML（Hyper Text Markup Language，超文本标记语言）通过使用很多标签来描述网

页。浏览器就是按照 HTML 标签的语义规则把 HTML 代码翻译成漂亮的网页的。

用户界面的美观对于软件系统来说是很重要的，CSS（Cascading Style Sheets，层叠样式表，也就是通常所说的样式表）就是增强用户页面的技术，确切地说，是一种美化网页的技术。通过使用 CSS，可以方便、灵活地设置网页中不同元素的外观属性，通过这些设置可以使网页的外观达到一个更高的水准。

JavaScript 脚本在浏览器端运行，可以完成一些与用户的互动，如验证用户输入的字符串中是否包含“@”符号，来判断用户输入的 E-mail 地址是否有效，不用发送到服务器端处理，从而可减轻网络和服务器的负担。

AJAX（Asynchronous JavaScript and XML，异步 JavaScript 和 XML）是指一种创建交互式网页应用的网页开发技术，其核心是 JavaScript 对象 XMLHttpRequest。XMLHttpRequest 是一种支持异步请求的技术，可在不重载页面的情况下与 Web 服务器交换数据。在 AJAX 技术没有出现以前，Web 应用程序最不能让用户满意的地方是其提交请求后漫长的等待时间，尤其是在大量用户同时访问服务器的时候，而且其单调笨拙的用户界面，也让用户感到遗憾。AJAX 可以使网页只刷新变动的部分，能够快速响应用户请求，为用户提供不因页面刷新而中断的连续用户体验，还有类似于 google 的自动补全功能，可使用户获得更佳的用户体验。DOJO 和 JQuery 是常用的 AJAX 框架，它们使 AJAX 开发更加轻松。

1.3.2 动态网页开发技术

Java 动态网页开发技术有 Servlet 和 JSP。

Servlet 程序实际上就是用 Java Servlet API 开发的 Java 程序。Servlet 具有 Java 程序所具有的优点：跨平台、安全。Servlet 的缺点在于它的页面显示和业务逻辑没有分离，编写难度较大。

JSP（Java Server Pages）是由 Sun Microsystems 公司倡导、许多公司参与一起建立的一种动态网页技术标准，是对 Servlet 的简化。在 HTML 文件中加入 Java 程序片段和 JSP 标签，就构成了 JSP 页面。虽然 JSP 具有页面显示和业务处理混杂的情况，但是 JavaBean 的出现，以及 JSTL 的使用，很好地解决了这个问题，所以 JSP 是比较常用的动态网页技术，我们在后面的章节中将会详细介绍该技术。

1.3.3 SSH 框架——Java 轻量级企业应用解决方案

Java EE 是为了企业应用而提供的分布式、高可靠性的解决方案。以前的 JavaEE 应用使用 EJB 作为核心，门槛高，入门难，开发成本和部署成本都很高，这大大限制了它的使用。Struts、Spring、Hibernate 这些轻量级解决方案（简称为 SSH 框架）的出现，改变了这个局面。

Struts 是 MVC 设计模式的一个优秀实现，是最早的 Java 开源框架之一，也是现在 Java Web 框架的事实标准。MVC 是一种流行的软件设计模式，它把系统分为 3 个部分：模型 Model、视图 View 与控制器 Controller，使数据与显示分离，提高系统的可扩展性及可维护性。Struts 定义了通用的 Controller，通过配置文件（如 struts.xml）隔离 Model 和 View，

以 Action 的概念对用户请求做了封装,使代码更清晰易读。Struts 还提供了自动将请求的数据填充到对象中以及页面标签等简化编程的工具。Struts 使开发大型 Web 项目成为可能。

Spring 也是一个开源框架,是为简化企业级应用系统开发而推出的,通过使用 Spring,可以用简单的 Java Bean 实现以前只能用 EJB 才能完成的任务。

Hibernate 是一个开放源代码的对象关系映射框架,它对 JDBC 进行了非常轻量级的对象封装,使得 Java 程序员可以随心所欲地使用对象编程思维来操纵数据库,大大简化了数据访问烦琐的重复性代码,提高了开发效率。类似的框架还有 ibatis。ibatis 和 Hibernate 的区别就在于程序员需要写完整 SQL 语句,这意味着可以通过高度优化的 SQL 语句(或存储过程)提高系统性能,当然前提是需要精通 SQL 优化技术。

1.4 Servlet

Servlet 是一种动态 Web 组件技术,Servlet 程序实际上就是用 Java Servlet API 开发的遵循某种规范的、在服务器端运行的 Java 程序,可以在支持 Java 的 Web 服务器上运行。

Web 服务器中加载和运行 Servlet 的模块叫 Servlet 容器或 Servlet 引擎。如图 1.4 所示,浏览器发送请求给 Web 服务器,如果请求的是静态页面,Web 服务器将直接返回该页面。如果请求的是 Servlet,则 Web 服务器转发请求给 Servlet 容器,Servlet 容器再调用相关 Servlet 处理该请求,动态产生页面代码返回。

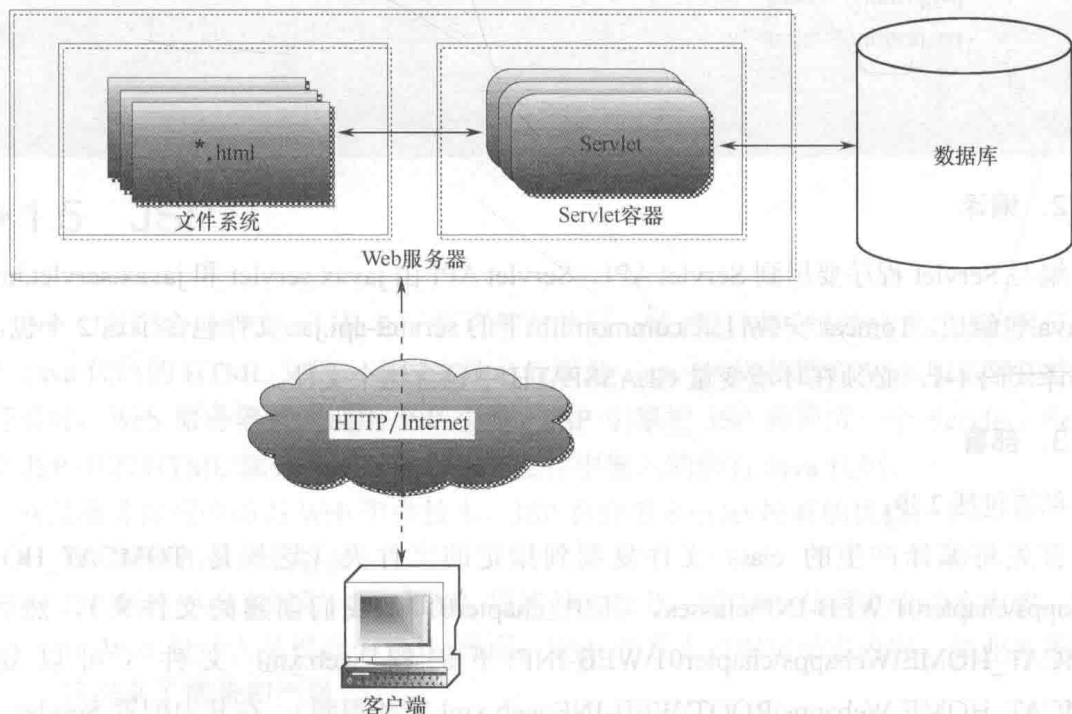


图 1.4 带有 Servlet 容器的 Web 应用程序