



21世纪普通高等教育应用型规划教材 · 计算机系列

JSP

网络程序设计

关东升 田登山 主编

JSP WANGLUO CHENGXU
SHEJI



北京邮电大学出版社
www.buptpress.com

21 世纪普通高等教育应用型规划教材·计算机系列

JSP 网络程序设计

关东升 田登山 主 编



北京邮电大学出版社
[www. buptpress. com](http://www.buptpress.com)

内 容 简 介

JSP 是 JavaEE 知识系统的核心内容之一,多年来 JavaEE 一直是企业级开发的主要技术,经过 10 多年的发展 JavaEE 已经很成熟了,JavaEE 也是相当的普及。为了能够在高校培养出更加适合企业需要的 Java 程序员,我们编写了这门 JSP 教材。

作为多年的培训讲师,我们了解企业的需求,也了解高校学生情况,为了能够培训出适合企业需要的学生,在以往授课过程中发现采用“项目驱动的、案例式教学模式”,学生接受起来很容易。因此,本书也采用了案例驱动模式展开讲解知识点,然后再展开介绍。案例驱动模式优势能够使学生学习了某一个知识点能够知道这个知识点在开发一个项目中所处的地位和作用等,当然,在具体的知识点讲解我又安排了许多的补充案例,充分的体现以“项目驱动的、案例式教学模式”,实践证明这是非常有效的教学模式。

在课程的编写过程中我们走访多家企业,我们所编制的知识点都是力求满足企业开发的需要,而非面面俱到面面不到方式。

本书适合于高校老师和学生学习使用 JSP 和 JavaEE,也适合于对于 Java 和 JavaEE 感兴趣的程序员、软件开发和系统设计人员。

图书在版编目(CIP)数据

JSP 网络程序设计/关东升,田登山主编. —北京:北京邮电大学出版社,2011.8

ISBN 978-7-5635-2699-4

I. ①J… II. ①关…②田… III. ①JAVA 语言—网页制作工具 IV. ①TP312②TP393.092

中国版本图书馆 CIP 数据核字(2011)第 152287 号

书 名: JSP 网络程序设计

主 编: 关东升 田登山

责任编辑: 满志文

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: 北京联兴华印刷厂

开 本: 787 mm×1 092 mm 1/16

印 张: 15

字 数: 372 千字

版 次: 2011 年 8 月第 1 版 2011 年 8 月第 1 次印刷

ISBN 978-7-5635-2699-4

定 价: 30.00 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 •

前 言

JSP 是 JavaEE 知识系统的核心内容之一,多年来 JavaEE 一直是企业级开发的主要技术,经过 10 多年的发展 JavaEE 已经很成熟了,JavaEE 也是相当的普及。为了能够在高校培养出更加适合企业需要的 Java 程序员,我们编写了这门 JSP 教材。下面是各章节介绍:

第 1 章——开篇综述

本章介绍了 JavaEE 的基本概念,JavaEE 包括的内容等,JavaEE 的组件和层次,以及 JavaEE 的优点。JavaEE 包括内容有 Servlet、JSP、JDBC、Java 事务 API(JTA)、Java 名字和目录接口(JNDI)、远程方法调用(RMI)、企业 JavaBean(EJB)、Java 消息服务(JMS)、XML、JavaMail 等。熟悉开发环境的搭建,如 JDK、Tomcat、Eclipse 和 MyEclipse 等工具的安装和使用。

第 2 章——图书管理系统分析与设计

本章介绍了图书管理系统的需求,当然这个需求是简化的。了解图书管理系统的设计。

第 3 章——XML 应用技术

本章介绍 XML 的基本概念、用途以及和 HTML 的区别。熟悉 XML 文件包括的内容,什么样是一个合法的 XML 文件。掌握通过 Java 解析 XML 文件。

第 4 章——数据持久层技术

本章介绍了数据持久化的概念,常用的数据持久化技术,包括:JDBC、EJB Entity、ORM (Hibernate)等。熟悉 JDBC 技术,对于不同的 JDBC 驱动程序决定了 Java 对数据库连接方式的不同,即不同的数据库加载的驱动程序是不同的。熟悉 JDBC 的几个常用接口:Driver、DriverManager、Connection、Statement 和 ResultSet 等接口。

第 5 章——JavaBean 组件技术

本章介绍了什么是 JavaBean 以及它具有的特性。牢记 JavaBean 的编写规范。熟悉 JavaBean 的属性以及几种常用的属性,简单属性和索引属性。

第 6 章——Web 技术基础

本章介绍了什么是表示层,了解表示层技术有哪些,掌握基于 Web 的表示层技术,熟悉 HTTP、HTML 和 JavaScript。

第 7 章——Servlet 技术

本章主要介绍了 Servlet 技术,包括 Servlet 的设计特点、Servlet 的生命周期。然后,介绍了 Servlet API,然后介绍了一个 Servlet 实例,并进行了分析。最后介绍了 Servlet 过滤器。

第 8 章——JSP 技术

本章主要介绍了 JSP 技术,包括 JSP 基本语法、JSP 内部对象,以及 MVC 设计模式。

第9章 利用 Struts 进行 Web 层开发

本章介绍了 Struts 框架基本原理,作为实现了 MVC 设计模式的 Struts 框架,它的各个层之间是如何相互作用的,介绍 Struts 框架中的模型组件。介绍 Struts 框架中的控制组件以及 Struts 控制层都有哪些组件。

第10章 利用 Hibernate 进行持久层开发

本章介绍了 Hibernate 在数据持久层的重要作用,能够通过 Hibernate 工具从数据库表结构生成 hbm.xml 文件以及业务领域对象,介绍 Hibernate 配置项目,介绍 Hibernate 持久化操作。熟悉在图书管理系统中如何利用 Hibernate 实现数据持久化的。

第11章 利用 Spring 构建松耦合应用系统

本章首先提出各层间依赖关系的弊端。然后,利用 Spring 建立松耦合图书管理系统,来展开 Spring 介绍。

第12章 AJAX 技术

本章首先介绍了一下 AJAX 技术,然后,介绍了 AJAX 基本原理,包括:创建 XMLHttpRequest 对象、建立连接、发送请求、制定回调方式等。最后,介绍了 AJAX 与服务器端程序 Servlet 通信。

参加本书编写、调试工作的人员还有田登山、赵志荣和贾云龙。本书的顺利出版,还要感谢 516dev(www.51work6.com)团队给予的大力支持和帮助。

由于时间仓促,书中难免存在不妥之处,请读者原谅,并提出宝贵意见。

作 者

目 录

第 1 章 开篇综述	1
1.1 JavaEE 的优势	1
1.1.1 保留现存的 IT 资产	2
1.1.2 高效的开发	2
1.1.3 可伸缩性	2
1.1.4 稳定的可用性	3
1.2 JavaEE 的组件	3
1.2.1 Servlet 与 JSP	3
1.2.2 JDBC	3
1.2.3 JNDI (Java Naming & Directory Interface)	4
1.2.4 RMI (Remote Method Invocation)	4
1.2.5 JMS (Java Message Service)	4
1.2.6 EJB	4
1.3 JavaEE 应用系统架构设计	5
1.3.1 表示层	5
1.3.2 服务层	6
1.3.3 数据持久层	6
1.3.4 企业信息系统层	6
1.4 开发环境的搭建	6
1.4.1 Tomcat 下载与安装	6
1.4.2 MyEclipse 下载与安装	9
第 2 章 图书管理系统分析与设计	15
2.1 系统需求	15
2.2 需求分析	15
2.2.1 图书管理系统的静态页面	15
2.2.2 系统用例	19
2.2.3 分析类	20
2.3 系统设计	21
2.3.1 系统类图	21
2.3.2 数据库物理模型	23



第3章 XML 应用技术	24
3.1 XML 概述	24
3.1.1 XML 的用途	24
3.1.2 XML 与 HTML 的区别	24
3.1.3 良好的格式	25
3.1.4 XML 文件需要验证(Validity)	25
3.2 XML 文件	26
3.2.1 XML 文件的结构	26
3.2.2 XML 文件的组成	26
3.2.3 XML 文件的声明	27
3.2.4 XML 文件的语法	28
3.3 在 Java 程序中访问 XML 文件	29
3.3.1 JAXP API 基础	29
3.3.2 DOM 对象模型	29
3.4 图书管理系统中的 XML 应用	34
第4章 数据持久层技术	35
4.1 数据持久层技术概述	35
4.1.1 Java 对象的序列化	35
4.1.2 Java 对象保存到数据库中	35
4.2 JDBC 技术	36
4.3 数据库驱动器程序	36
4.3.1 驱动程序分类	36
4.3.2 装载方法	39
4.3.3 DriverManager	39
4.4 JDBC 中的接口	41
4.4.1 Connection 接口	41
4.4.2 Statement 接口	41
4.4.3 ResultSet 接口	43
4.5 案例分析	44
4.5.1 JDBC-ODBC 桥方式连接数据库	44
4.5.2 MySQL 数据库访问	45
4.6 图书管理系统中的数据持久层技术	49
4.6.1 图书管理系统中 DAO 设计模式	49
4.6.2 图书管理系统中 JDBC 硬编码问题	57
第5章 JavaBean 组件技术	59
5.1 JavaBean 组件	59



5.1.1	什么是 JavaBean	59
5.1.2	JavaBean 编写规范	59
5.1.3	JavaBean 的属性	60
5.1.4	JavaBean 的方法访问	61
5.2	图书管理系统中的 JavaBean	63
5.2.1	JavaBean 分类	63
5.2.2	回顾设计类图	64
第 6 章	Web 技术基础	66
6.1	表示层技术概述	66
6.2	HTTP 协议	66
6.2.1	HTTP 协议的主要特点	67
6.2.2	HTTP 协议中几个概念	67
6.2.3	HTTP 协议的基本原理	67
6.3	HTML 超文本标识语言	69
6.3.1	HTML	69
6.3.2	HTML 中最基本的标签	71
6.3.3	属性	71
6.3.4	<HTML> 标签	72
6.3.5	<HEAD> 标签	72
6.3.6	<BODY> 标签	73
6.3.7	<TITLE> 标签	74
6.3.8	<Hn> 标签	76
6.3.9	<HR> 标签	77
6.3.10	<P> 标签	78
6.3.11	 标签	80
6.3.12	 标签	81
6.3.13	HTML 的图片	83
6.3.14	<A> 标签	85
6.3.15	HTML 的表格	86
6.3.16	HTML 的框架	90
6.3.17	表单	93
6.4	JavaScript 脚本语言	97
6.4.1	信息输出	98
6.4.2	JavaScript 的数据类型	100
6.4.3	变量的赋值	100
6.4.4	JavaScript 中的注释	103
6.4.5	函数	103

第 7 章 Servlet 技术	106
7.1 Servlet 概述	106
7.1.1 Servlet 的技术特点	106
7.1.2 Servlet 的生命周期	107
7.1.3 Servlet 的功能	108
7.2 Servlet API	108
7.2.1 Servlet 接口和相关类	109
7.2.2 Servlet 请求和响应	110
7.2.3 HttpSession	111
7.3 Servlet 实例分析	111
7.3.1 使用 HttpServlet 处理客户请求	111
7.3.2 Web 应用系统的目录结构	115
7.3.3 Servlet 在 Web.xml 文件中的配置	115
7.4 Servlet 过滤器	117
7.4.1 Servlet 过滤器机制介绍	117
7.4.2 Servlet 过滤器接口介绍	118
7.4.3 Servlet 过滤器实例分析	119
第 8 章 JSP 技术	123
8.1 JSP 基本语法	125
8.1.1 模板元素	125
8.1.2 注释	125
8.1.3 脚本元素	127
8.1.4 指令元素	130
8.1.5 动作元素	134
8.2 内部对象	144
8.2.1 request 对象	144
8.2.2 response 对象	150
8.2.3 out 对象	151
8.2.4 session 对象	152
8.3 MVC 设计模式	155
8.3.1 表示层和业务层分离问题	156
8.3.2 JSP 的两种体系结构	159
8.3.3 MVC 设计模式说明	160
第 9 章 利用 Struts 进行 Web 层开发	163
9.1 第一个 Struts 工程	163
9.2 Struts 框架介绍	178



9.2.1 概述	178
9.2.2 Struts 主要组件	179
9.3 Struts 标签库	179
9.3.1 Struts 标签库分类	179
9.3.2 使用标签库的步骤	181
9.4 Struts 的输入验证	190
9.4.1 实现 validate 方法	190
9.4.2 错误消息国际化	192
9.4.3 与错误消息有关的 JSP 部分	194
9.4.4 Action 中的验证	195
第 10 章 利用 Hibernate 进行持久层开发	197
10.1 Hibernate 概述	197
10.1.1 Hibernate 体系结构	197
10.1.2 第一个 Hibernate 例子	198
10.2 操作持久化数据	204
10.2.1 装载对象	205
10.2.2 查询对象	206
10.2.3 更新对象	208
10.2.4 删除对象	209
第 11 章 利用 Spring 建立松耦合应用系统	210
11.1 各层间的依赖关系	210
11.2 利用 Spring 建立松耦合图书管理系统	211
第 12 章 AJAX 技术	220
12.1 AJAX 介绍	221
12.1.1 AJAX 技术本质	221
12.1.2 HTTP 请求和 XMLHttpRequest 对象	221
12.2 AJAX 原理	222
12.2.1 创建 XMLHttpRequest 对象	223
12.2.2 建立请求	224
12.2.3 发送请求	225
12.2.4 指定回调方法	225
12.2.5 服务器的处理 Servlet	227
参考文献	229

第 1 章 开篇综述

JavaEE(Java Platform, Enterprise Edition)是一个开发分布式企业级应用的规范。它提供了一个多层次的分布式应用模型和一系列开发技术规范。首先应该明白 J2EE 是一个标准,而不是产品。多层次分布式应用模型是指根据功能把应用逻辑分成若干个层次,每个层次支持相应的服务器和组件,组件在分布式服务器的组件容器中运行(如 Servlet 组件在 Servlet 容器上运行,EJB 组件在 EJB 容器上运行),容器间通过相关的协议进行通信,实现组件间的相互调用。遵从这个规范的开发将得到行业的广泛支持,使企业级应用的开发变得简单、快速。



提示

在 JavaEE 的早期版本中,通常简称为 J2EE 的 J 代表 Java,数字 2 代表 Java 的第 2 版, E 代表企业级版本(Enterprise Edition)。

Java 不仅是编程语言,还是一个开发平台,Java 技术给程序员提供了许多工具:编译器、解释器、文档生成器和文件打包工具等。同时 Java 还是一个程序发布平台,有两种主要的“发布环境”,首先 Java 运行时环境(Java Runtime Environment, JRE)包含了完整的类文件包,其次许多主要的浏览器都提供了 Java 解释器和运行时环境。目前原 Sun 公司(现在被 Oracle 公司收购)把 Java 平台划分成 JavaEE、JavaSE 和 JavaME 三个平台,早期 Java 版本是 Java2,因此现在很多人还习惯于称之为 J2EE、J2SE 和 J2ME,针对不同的市场目标和设备进行定位。J2EE 是 Java2 Enterprise Edition,主要目的是为企业计算提供一个应用服务器的运行和开发平台。J2EE 本身是一个开放的标准,任何软件厂商都可以推出自己的符合 J2EE 标准的产品,使用户可以有多种选择。IBM、Oracle、BEA、HP 等 29 家已经推出了自己的产品,其中尤以 BEA 公司的 Weblogic 产品和 IBM 公司的 Websphere 最为著名。J2EE 将逐步发展成为可以与微软公司的 .NET 战略相对抗的网络计算平台。J2SE 是 Java2 Standard Edition,主要目的是为台式机和工作站提供一个开发和运行的平台。我们在学习 Java 的过程中,主要是采用 J2SE 来进行开发。J2ME 是 Java2 Micro Edition,主要是面向消费电子产品,为消费电子产品提供一个 Java 的运行平台,使得 Java 程序能够在手机、机顶盒、PDA 等产品上运行。

1.1 JavaEE 的优势

JavaEE 是一种利用 Java 平台简化企业解决方案的开发、部署和管理相关复杂问题的体系结构。JavaEE 技术的基础就是核心 Java 平台或 Java 平台的标准版(JavaSE),JavaEE 不仅巩固了标准版中的许多优点,例如“编写一次、随处运行”的特性、方便存取数据库的 JDBC API、CORBA 技术,以及能够在 Internet 应用中保护数据的安全模式等,同时还提供



了对 EJB(Enterprise JavaBean)、Java Servlet API、JSP(Java Server Page)及 XML 技术的全面支持。其最终目的就是要成为一个能够使企业开发者大幅缩短投放市场时间的体系结构。

J2EE 体系结构提供中间层集成框架用来满足无须太多费用而又需要高可用性、高可靠性及可扩展性的应用的需求。通过提供统一的开发平台, J2EE 降低了开发多层应用的费用和复杂性, 同时提供对现有应用程序集成强有力的支持, 完全支持 Enterprise JavaBeans 有良好的向导支持打包和部署应用, 添加目录支持, 增强了安全机制, 提高了性能。J2EE 目前得到了广泛的应用, 其优势主要是为搭建具有可伸缩性、灵活性、易维护性的业务系统提供了良好的机制。下面对其优势做一详细叙述。

1.1.1 保留现存的 IT 资产

由于企业必须适应新的商业需求, 利用已有的企业信息系统方面的投资而不是重新制定全盘方案就变得很重要。这样, 一个以渐进的(而不是激进的、全盘否定的)方式建立在已有系统之上的服务器端平台机制是公司所需求的。JavaEE 架构可以充分利用用户原有的投资, 如一些公司使用的 BEA Tuxedo、IBM CICS、IBM Encina、Inprise VisiBroker 及 Netscape Application Server。这之所以成为可能, 是因为 JavaEE 拥有广泛的业界支持和一些重要的“企业计算”领域供应商的参与。每一个供应商都对现有的客户提供了不用废弃已有投资即可进入可移植的 JavaEE 领域的升级途径。由于基于 JavaEE 平台的产品几乎能够在任何操作系统和硬件配置上运行, 现有的操作系统和硬件也能被保留使用。

1.1.2 高效的开发

JavaEE 允许公司把一些通用的、很烦琐的服务器端任务交给中间件供应商去完成, 这样开发人员可以集中精力在如何创建业务逻辑上, 相应地缩短了开发时间。高级中间件供应商提供以下这些复杂的中间件服务:

(1) 状态管理服务: 让开发人员写更少的代码, 不用关心如何管理状态, 这样能够更快地完成程序开发。

(2) 持续性服务: 让开发人员不用对数据访问逻辑进行编码就能编写应用程序, 能生成更轻巧且与数据库无关的应用程序, 这种应用程序更易于开发与维护。

(3) 分布式共享数据对象缓存服务: 让开发人员编制高性能的系统, 极大地提高了整体部署的伸缩性。

1.1.3 可伸缩性

企业必须选择一种服务器端平台, 这种平台应能提供极佳的可伸缩性去满足那些在他们系统上进行商业运作的大批新客户。基于 JavaEE 平台的应用程序可被部署到各种操作系统上。例如可被部署到高端 UNIX 与大型机系统, 这种系统单机可支持 64~256 个处理器。JavaEE 领域的供应商提供了更为广泛的负载平衡策略, 能消除系统中的瓶颈, 允许多台服务器集成部署。这种部署可达数千个处理器, 实现可高度伸缩的系统, 满足未来商业应用的需要。



1.1.4 稳定的可用性

一个服务器端平台必须能全天候运转以满足公司客户、合作伙伴的需要。因为 Internet 是全球化的、无处不在的,即使在夜间按计划停机也可能造成严重损失。若是意外停机,那会有灾难性后果。JavaEE 部署到可靠的操作环境中,支持长期的可用性。一些 J2EE 部署在 Windows 环境中,客户也可选择健壮性能更好的操作系统,如 Sun Solaris、IBM OS/390。正是具有了以上优点,JavaEE 才会受到企业客户及软件开发者的青睐,成为当前软件开发领域的一大技术主流。

1.2 JavaEE 的组件

JavaEE 是一个技术架构,也可以视为一个技术体系,它包含了许多内容,这也是其复杂性之所在。JavaEE 主要包括了表 1-1 所示的技术。

表 1-1 JavaEE 主要技术

技术点	说 明	技术点	说 明
Servlet	表示逻辑	Java 名字和目录接口(JNDI)	对象名字的中央注册
JSP	表示逻辑	远程方法调用(RMI)	RMI 对象的分布式执行
Java 数据库连接(JDBC)	数据库访问	企业 JavaBean(EJB)	实现业务逻辑
Java 事务 API(JTA)	事务管理	Java 消息服务(JMS)	协调分布式执行

1.2.1 Servlet 与 JSP

Servlet 是一种服务器端的技术,用来实现表示逻辑。它接收来自 Web 浏览器的 HTTP 请求信息,进行处理后返回 HTTP 响应结果信息。由于 Servlet 是用 Java 编写的,可移植性较强,可以从一个平台移植到另一个平台。WebLogic Server 中的 Servlet 采用了请求/响应模型。当来自客户机的请求到达 Servlet 引擎时,WebLogic Server 将调用适当的 Servlet 进行处理,然后把响应信息返回给客户机。目前常用的 Servlet 类型是 HTTP Servlet,该类型的 Servlet 是专门为采用 HTTP 协议请求服务的。WebLogic Server 仅支持 HTTP Servlet。

JSP(Java Server Page)技术可以提供与 Servlet 相同的功能,但是开发起来比 Servlet 要简便。JSP 可以帮助 Web 开发者使用相对较少的代码创建动态的网页。JSP 的优点是它的简洁性,Web 设计者即使不懂得编程,也可以使用 JSP 创建动态的网页。JSP 是 HTML 代码和 Java 代码的混合。当第一次请求一个 JSP 页面时,应用服务器需要首先把该 JSP 页面编译为一个 Servlet,为请求提供服务。在客户请求 JSP 页面的时候,服务器就会处理 Java 代码,然后返回 HTML 页面给浏览器。

1.2.2 JDBC

JDBC(Java Data Base Connectivity)为 Java 应用程序访问数据库提供了一个统一的接口。通过 JDBC 可以完成建立到数据库的连接、执行 SQL 语句等操作。

通过 JDBC 访问数据库需要有相应数据库的 JDBC 驱动程序(通常由数据库厂商提供),JDBC 驱动程序负责将 JDBC 操作转换为特定的数据库操作。WebLogic Server 提供了大量的 JDBC 驱动程序,例如针对 Oracle、SQL Server、Sybase 等数据库的 JDBC 驱动。

1.2.3 JNDI (Java Naming & Directory Interface)

JNDI 即 Java 的命名和目录接口。JNDI 是 Java 定义的一种标准的注册机制,提供名字和目录服务。在 J2EE 应用中,JNDI 用于管理应用中用到的核心组件和对象。当需要使用某一个组件时,JNDI 可以提供查找该组件的方法。

JNDI 提供了一组 API 用来访问命名和目录服务。它提供一个相容的模式访问和操作企业大范围的资源,例如一个应用服务器中的 DNS、LDAP、本地文件系统或者对象。

命名服务是一种应用程序,该程序包含一个对象集合或对象引用的集合,还对每个对象关联了一个对用户友好的名称。目录服务只是命名服务所提供的的一个扩展功能。

WebLogic Server 提供了 JNDI 服务。开发人员可以存储、修改和删除 WebLogic Server JNDI 中的 Java 组件和资源。在使用 WebLogic Server 集群的情形下,JNDI 也用做整个集群中的共享名字服务。

1.2.4 RMI (Remote Method Invocation)

RMI 即远程方法调用。RMI 是 Java 应用程序访问远程 Java 对象的一种标准。利用 RMI,Java 应用程序可以访问位于其他进程甚至其他计算机上的对象。对于应用程序来说,RMI 使得远程对象如同本地对象一样。RMI 类似于早期的 CORBA(Common Object Request Broker Architecture,公共对象请求代理体系结构,通用对象请求代理体系结构),区别是 CORBA 是异构环境下对象调用,即 Java 调用 C++ 等,而 RMI 是实现 Java 对象之间的调用。

1.2.5 JMS (Java Message Service)

JMS(Java 消息服务)提供了一组标准的 Java API,用于企业级的消息处理。通过 JMS,可以在 Java 程序之间发送和接收消息。WebLogic Server 完全实现了 JMS 标准。JMS 提供两种消息处理方式:队列(Queue)模型和基于主题的发布/订阅模式。

队列模型是将 JMS 消息放到 JMS 队列中,然后接收方到 JMS 队列中去取消息。发布/订阅模式是将某一主题的 JMS 消息发给该主题的所有订阅者,与生活中订阅报刊有点类似。

1.2.6 EJB

JavaEE 中一个引人注目的技术就是 EJB。它提供了一个架构,开发和配置到客户端的分布式业务逻辑,因此可以明显减少开发扩展性高度复杂的企业应用的难度。EJB 规范定义了 EJB 组件应该如何及何时与它们的容器交互。由 EJB 容器负责提供普通的服务,例如目录服务、事务管理、安全、资源池和容错。EJB 3.0 规范定义了以下几种 EJB:

1. 会话 Bean

会话 Bean 在 JavaEE 应用程序中被用来完成一些服务器端的业务操作,例如访问数据库、调用其他 EJB 组件。在一个基于 JavaEE 规范的电子商务系统中会话 Bean 可能会充当以下角色:计价器、信用卡验证器、商品搜索引擎。总之,会话 Bean 代表的是“动作”。会话 Bean 分为



两类:有状态会话 Bean 和无状态会话 Bean。两者的区别在于是否保存客户端的状态信息。

2. 实体 Bean

实体 Bean 代表持久保存的数据,典型的是存储在数据库中的数据,因此在服务器崩溃后数据仍然存在。多个客户端可以使用 EJB 表示同样的数据。例如在一个电子商务系统中,实体 Bean 可能被用来代表产品、雇员、信用卡、订单等数据对象。

3. 消息驱动 Bean

结合了会话 Bean 和 JMS 的消息监听器的特性,允许一个业务层组件异步接收 JMS 消息。虽然以下几种 EJB 各有不同,不过所有的 EJB 也有不少相同的地方。例如,它们都有一个 home 接口,用来定义一个客户如何被创建和消除;一个远程的接口,定义客户端可以调用哪些 Bean 上的方法;以及一个实现主商业逻辑的 Bean 类。

1.3 JavaEE 应用系统架构设计

我们设计和开发一个企业级应用软件系统时候离不开系统架构设计。软件设计的原则是为了提高软件系统的可复用性和可扩展性,系统结构设计是软件设计的基础,我们采用的手段是系统的分层次。这是一种逻辑上的划分不是物理上的划分,也就是这些层次可以是在一台计算机上当然也可以分布到在多台计算机上。这些层次之间是松耦合的,层次的内部是高内聚的。因此,降低耦合是软件设计的目标,能够设计出低耦合的系统,就意味着系统具有可复用性和可扩展性了。如图 1-1 所示是一般企业级软件系统体系结构图。

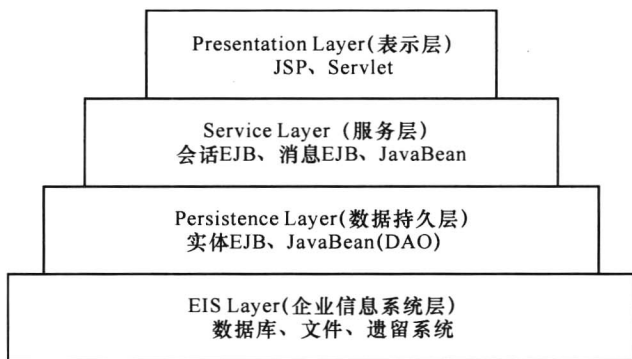


图 1-1 JavaEE 应用系统架构设计

1.3.1 表示层

表示层是用户与系统交互的组件集合,用户通过这一层向系统提交请求或发出指令,系统通过这一层接收用户请求或指令,然后,将指令消化吸收后调用下一层,再将调用的结果展现到这一层。表示层应该是轻薄的不应该具有业务逻辑,如果我们的系统是一个 Web 系统的话,我们使用的技术 JSP、Servlet 等,可以使用 MVC 框架例如 Struts 等框架技术,这时我们可以将表示层称为 web 层更准确一点,因为在 MVC 框架中 JSP 作为表示层,Servlet 作为控制层。但是不论采用什么技术这层都应该是轻薄的,也就是说无论使用 JSP、Servlet 作为表示层还是使用 Swing 作为表示层都应该是轻薄的。



1.3.2 服务层

服务层是系统的核心业务处理层,负责接收表示层的指令和数据,消化吸收后,进行组织业务逻辑的处理,并将结果返回给表示层。为了降低表示层和服务层的耦合问题,我们会在两层之间引入接口,然后在运行期注入依赖关系。服务层一般使用的技术有会话 EJB、消息 EJB 和普通 JavaBean,这里 JavaBean 一般没有属性只有业务方法。

1.3.3 数据持久层

数据持久层是服务层用于访问数据库层,从设计规范上讲为了降低耦合度,服务层不应该具有访问数据库的代码,访问数据库的代码应该放到数据持久层中,在数据持久层中一般是通过 DAO(数据访问对象)访问数据库的,也是为了降低耦合度,DAO-被设计为接口,DAO 实现部分中通过 JDBC 也可以使用一些持久层框架如 Hibernate 等。除了使用 DAO 以外我们也可以使用 EJB 中的实体 EJB。

1.3.4 企业信息系统层

企业信息系统层(EIS)是系统的数据来源,可以是数据库、文件和遗留系统。数据库不用过多的介绍了,多数情况下指的是数据库,而且一般都是关系型数据库;文件是指数据储存在文件中的,一般用于小型系统数据量很小或没有数据库环境;遗留系统是指数据来源于别的系统,例如一个统计报表系统数据来源于原来的业务系统。

1.4 开发环境的搭建

J2EE 的开发是基于 Java 开发之上的,集成开发工具一般还是使用 Eclipse,由于 Eclipse 只有开发一般的 Java 程序,开发不了 JavaEE,如不能建立 JSP 等,也没有 JavaEE 的运行环境,要开发 JavaEE 必须安装插件,在众多开发 JavaEE 的插件中备受青睐的有 MyEclipse 和 Lombok 等插件。MyEclipse 是一个付费插件,Lombok 目前还是免费插件。本书主要使用 MyEclipse 插件,Web 服务器采用 Tomcat。

1.4.1 Tomcat 下载与安装

Tomcat 是一个 Web 服务器,在 J2EE 的 Web 应用系统中非常重要,它负责 JSP、Servlet 解释与运行,负责接收 HTTP 请求和应答。



提示

如果我们使用的 MyEclipse 版本比较新,MyEclipse 会自带一个 Tomcat,这个自带的 Tomcat 的版本要看 MyEclipse 版本而定。

1. 下载

本书使用 Tomcat6.0X 版本,下载地址:<http://tomcat.apache.org/download-60.cgi>,如图 1-2 所示,为了方便起见可以下载 Windows 安装版本(32-bit/64-bit Windows Service Installer),安装版本会自带配置环境变量,如果下载的是压缩版本,需要注意计算机的操作

系统是 32 位还是 64 位,根据操作系统位数选择 64-bit Windows zip 还是 32-bit Windows zip。

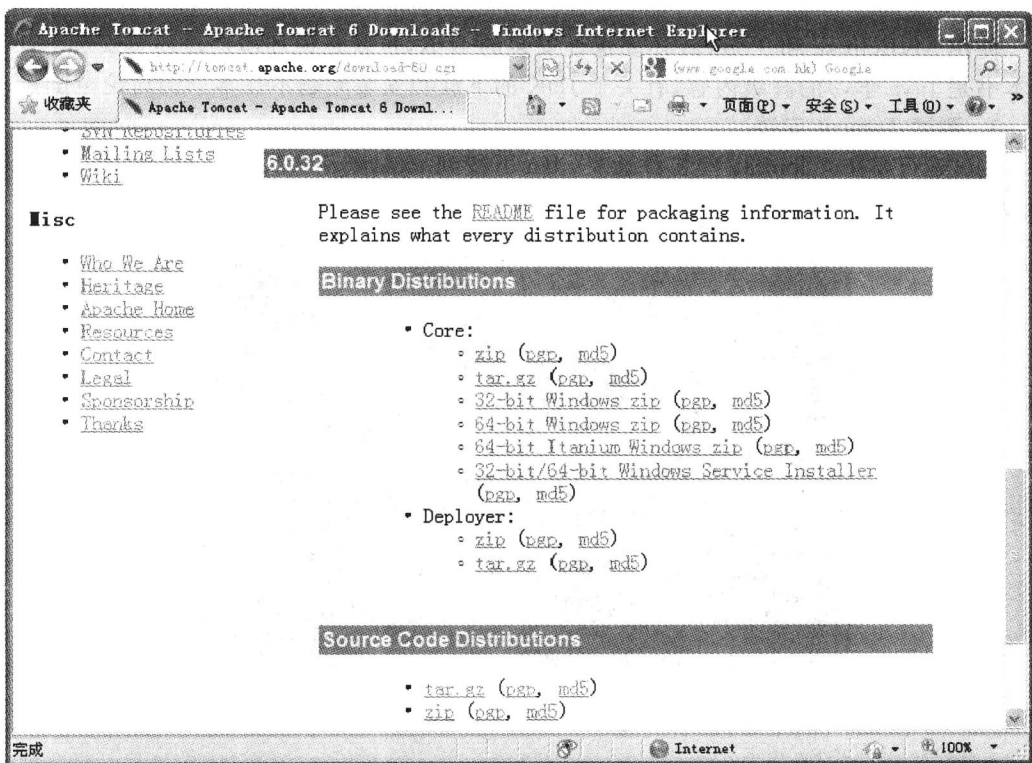


图 1-2 下载 Tomcat

2. 安装

由于下载的是 Windows 安装版本,安装过程比较简单,但是在设置端口和管理员密码的时候需要注意一下。如图 1-3 所示,这里我们默认使用 8080 端口,管理员密码根据习惯自己设置就可以了。

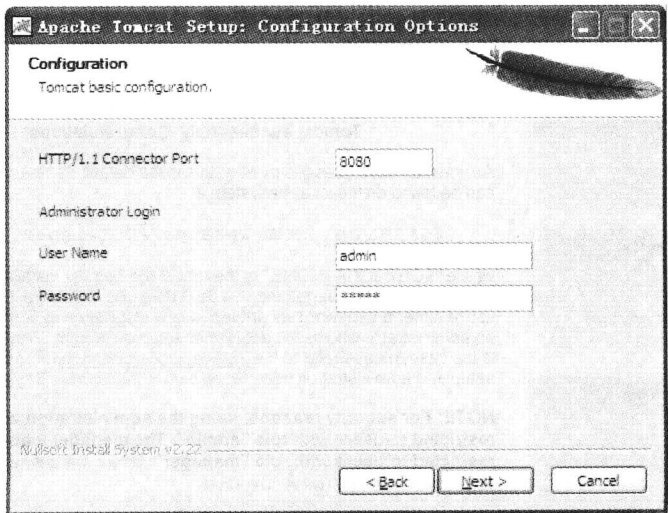


图 1-3 设置端口和管理员密码