

Struts+Hibernate

开发Web应用 ——从基础到实践

| 吴明晖 吴超 编著

介绍如何利用Struts和Hibernate协同开发基于J2EE的Web应用，在分别介绍Struts和Hibernate的基础上，通过大量实例展示实际开发过程

介绍Eclipse开发环境的使用以及安装配置过程，一步步帮助读者上手

含有许多完整的Web应用实例，读者不仅可以通过它们掌握Struts和Hibernate的开发知识，还可以通过改动这些实例，得到可实际应用的Web应用代码

介绍如何进行J2EE Web应用的测试，帮助读者更好地融入现今的工业开发环境中

CD-ROM 附光盘



Struts+Hibernate

开发Web应用 ——从基础到实践

| 吴明晖 吴超 编著

人民邮电出版社
北京

图书在版编目 (CIP) 数据

Struts+Hibernate 开发 Web 应用: 从基础到实践 / 吴明晖, 吴超编著.

—北京: 人民邮电出版社, 2007.5

ISBN 978-7-115-15941-0

I. S... II. ①吴...②吴... III. ①软件工具—程序设计—教材②JAVA 语言—程序设计—教材 IV. TP311.56 TP312

中国版本图书馆 CIP 数据核字 (2007) 第 030420 号

内 容 提 要

本书介绍如何利用 Struts 和 Hibernate 框架开发 J2EE 平台下的 Web 应用, 分别讲述了 Struts 和 Hibernate 的基本知识, 以及如何将两者结合起来进行开发。本书主要分成两个部分: 第一部分为理论知识部分, 主要讲述 Struts 和 Hibernate 协同开发的技术细节。第二部分为实例部分, 利用 Struts 和 Hibernate 开发了一些有实际应用价值的 Web 应用实例, 包括 RSS 阅读器、用户管理系统、网上论坛、考试成绩维护系统等。最后还对如何进行测试进行了详细的介绍。

本书适合有 Java 基础, 但对 Struts 和 Hibernate 不甚了解的读者阅读, 可以帮助读者快速掌握利用这两个开源框架开发 J2EE 下 Web 应用的方法。

Struts+Hibernate 开发 Web 应用——从基础到实践

- ◆ 编 著 吴明晖 吴 超
责任编辑 刘 浩
- ◆ 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@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京密云春雷印刷厂印刷
新华书店总店北京发行所经销
- ◆ 开本: 787×1092 1/16
印张: 24.75
字数: 604 千字 2007 年 5 月第 1 版
印数: 1—4 000 册 2007 年 5 月北京第 1 次印刷

ISBN 978-7-115-15941-0/TP

定价: 45.00 元 (附光盘)

读者服务热线: (010)67132692 印装质量热线: (010)67129223

前言

J2EE 是目前世界上开发 Web 应用（特别是企业级 Web 应用）最流行的平台之一。经过多年的工业实践之后，在 J2EE 平台下产生了众多优秀的技术和框架，其中大多数为开源技术和框架。在实际软件开发中运用这些框架，大大降低了 J2EE 开发的复杂度和难度，也降低了开发成本。

本书介绍了 Struts 和 Hibernate 这两个应用极为广泛的 J2EE 开源框架。利用这两个强大的框架可以轻松地实现高效且符合工业要求标准的 J2EE Web 应用。

Struts 是 Apache 基金会 Jakarta 项目下的一个开源项目，被公认为是开发基于 MVC 模式 Web 应用的最经典的框架。Struts 框架将设计模式中“分离显示逻辑与业务逻辑”的能力发挥得淋漓尽致。Struts 遵守了 J2EE 的 Servlet、JSP 等规范，并且根据 J2EE 的特点做了相应的变化和扩展，是 J2EE 体系架构的一种轻量级实现。Struts 框架凭借其清晰性和灵活性，被越来越多的 Web 应用项目所采用，并成为当前应用最为广泛的轻量级 Web 开发框架。

Hibernate 是一个对象关系映射（Object-Relational Mapping）的开源框架，是基于 Java 的非常优秀的一个对象持久化开发框架。它根据 O/R（对象/关系）映射技术思想，对 JDBC 进行了轻量级封装，使开发人员可以使用一种面向对象的方式来操纵数据库。Hibernate 可以应用在任何使用 JDBC 的场合，其中最重要的是，Hibernate 可以在 J2EE Web 应用中完成数据持久化工作。

目前，基于 Struts 和 Hibernate 的项目越来越多。把 Struts 和 Hibernate 这两个框架结合起来体现了一种小而便捷的 J2EE 架构，很好地实现了 MVC 的设计模式。甚至有人认为 Struts+Hibernate 已是 J2EE 的一种标准开发模式。本书就是要讲述如何实现基于 Struts 和 Hibernate 的 J2EE 开发，在介绍 Struts 和 Hibernate 的各自技术的基础上，将两者的特点结合起来，帮助读者了解基于 Struts 和 Hibernate 的开发原理和细节。

本书按照理论到实践、从浅到深的顺序来组织安排内容，全书主要分为两部分。

1. 理论基础部分

第 1 章介绍 J2EE 平台的基础知识。首先介绍了 Web 应用的概念，以及 J2EE Web 应用的基本组成，然后向读者介绍 J2EE 平台出现的原因及其特性和优点，接着讲述 J2EE 平台下的主要技术规范，引导读者把握 J2EE 平台的整体特点，最后介绍 J2EE 服务器的内容以及参与 J2EE 平台开发的各种角色。

第 2 章概要介绍 Struts 的基本内容。首先帮助读者理解 Struts 的必需预备知识，包括 Web 应用的生命周期、Servlet、JSP、JavaBean、XML 等，接下去介绍 Struts 框架的特性，最后描述 MVC 设计模式的思想，以及 Struts 框架是如何实现 MVC 模式的。

第 3 章概要介绍 Hibernate 框架。首先介绍了学习 Hibernate 所必须的预备知识，包括 JDBC 和数据持久化的概念，然后介绍 Hibernate 出现的原因、与 EJB 等技术的对比、Hibernate 编程的主要组成部分、Hibernate 使用的主要类和配置方法等。

第 4 章利用两个实例将 Struts 和 Hibernate 的内容结合起来,帮助读者理解 Struts 和 Hibernate 的基本理论知识,并通过实践来掌握 Struts 和 Hibernate 开发从配置到编程的全过程。

第 5 章介绍 Struts 的配置。包括如何配置 Web.xml 以使用 Struts、Struts-config.xml 的配置、多配置文件的使用等内容。

第 6 章介绍 Struts 的组件。首先介绍 Model 组件、View 组件和 Controller 组件,接着讲解 Struts 的 4 个核心组件,并具体阐述 Struts 的使用机制。

第 7 章介绍 Struts 标签。首先介绍标签库的概念和使用方法,然后重点介绍 Struts 的各个标签库,包括 Html 标签库、Bean 标签库、Logic 标签库和 Nested 标签库,最后介绍如何编写自定义标签。

第 8 章介绍 Struts 的一些常用插件和异常。首先介绍利用 Tiles 框架来提高页面元素的可重用性,然后介绍用于输入验证的 Validator 框架,接着介绍 Struts 的全局异常和局部异常等信息。

第 9 章介绍 Hibernate 的映射关联方式,包括一对一关联、一对多关联和多对多关联。

第 10 章介绍如何使用 Hibernate 进行数据操作。介绍了 3 种数据操作方式: HQL、Criteria Query 以及 SQL 查询。

第 11 章介绍 Hibernate 的其他知识。包括自动代码生成工具、事务与并发和延时加载等。

2. 综合实例部分

第 12 章中开发了与 RSS 阅读器有关的 3 个实例。第一个实例纯粹采用 Struts 框架开发简单的 RSS 在线阅读器,第二个实例纯粹采用 Hibernate 框架开发 RSS 的自动更新器,最后的一个实例将两者结合起来,综合利用 Struts 和 Hibernate 开发了一个带数据库的 RSS 在线阅读器。

第 13 章利用 Struts 和 Hibernate 框架开发了一个用户管理系统,首先帮助读者设计系统的功能、流程和主要模块,接着分别实现从数据库到前台页面的开发过程。

第 14 章开发一个简单的网上论坛系统,它可以从一个很简单的模型复杂细化到一个庞大的系统,此实例可以帮助读者理解 Struts 和 Hibernate 开发 Web 应用的一些关键技术。

第 15 章介绍了一个稍复杂的 Web 应用系统。这个系统维护一个考试成绩的数据库,通过本章的学习可以深入了解到 Struts 和 Hibernate 的一些技术特性。

本书适合于有 Java 编程基础的读者阅读,如果读者有 J2EE 的开发经验,则可以跳过第 1 章的内容。如果读者对 Struts 或 Hibernate 其中一个框架有所了解的话,可以直接阅读另一个框架的内容,并通过实例快速掌握两者结合的方法。

本书本着易学实用的原则,一步步引导读者轻松学习利用 Struts 和 Hibernate 的 J2EE 开发,相信读者在读完本书后一定会有所收益。

本书由浙江大学城市学院吴明晖和吴超主编,参加资料整理的还有柯海丰、侯宏伦、沈明伟、钟晴江、姚慧、张琦、俞雪永、高建礼、胡戥恒、章欣、杨武飞、丁峰等,在此表示感谢。由于时间紧迫和水平有限,书中难免存在不足,请广大读者批评指正(可发电子函件至: book_better@sohu.com)。

编者

2007.3

目 录

第 1 章 Web 应用程序与 J2EE 平台	1	3.1.2 持久化	38
1.1 Web 应用程序	1	3.2 O/R Mapping	38
1.2 J2EE 平台的产生	3	3.3 Hibernate 介绍	39
1.3 J2EE 平台特性	4	3.3.1 Hibernate 的背景和特性 ..	39
1.3.1 作为企业应用的特性	4	3.3.2 与 EJB 等其他技术的 比较	40
1.3.2 J2EE 的多层结构	6	3.3.3 Hibernate 的主要组成 ..	41
1.3.3 J2EE 的结构变形	7	3.3.4 Hibernate 的 Configuration、 SessionFactory 和 Session ..	47
1.4 J2EE 平台简介	9	3.3.5 Hibernate 的配置	49
1.4.1 J2EE 主要组成技术	10	第 4 章 Struts 和 Hibernate 的环境 配置与应用小实例	53
1.4.2 J2EE 容器	11	4.1 Struts 和 Hibernate 的开发 环境配置	53
1.5 J2EE 服务器	13	4.1.1 数据库的安装和管理	53
1.6 参与 J2EE 平台开发的各种 角色	13	4.1.2 Web Server 的安装和 管理	56
第 2 章 初识 Struts	15	4.1.3 Hibernate 的安装	57
2.1 Struts 预备知识	15	4.1.4 Struts 的安装	58
2.1.1 Web 应用程序的生命 周期	15	4.1.5 开发工具的选择	58
2.1.2 Java Servlet	15	4.1.6 配置 Eclipse	61
2.1.3 JavaServer Pages (JSP) ..	17	4.2 实例一: 用户密码验证和登录 ..	67
2.1.4 JavaBean	23	4.2.1 总体设计	67
2.1.5 XML 基础	26	4.2.2 具体实现	68
2.2 Struts 框架	28	4.2.3 实例小结	86
2.2.1 Struts 框架的历史与 目前的商业应用	28	4.3 实例二: 用户密码修改	86
2.2.2 Struts 框架的特性和 优点	29	4.3.1 总体设计	86
2.2.3 与 JSTL、JSF 等其他技术 框架的比较	29	4.3.2 具体实现	86
2.3 Struts 的 MVC 构架	34	4.3.3 运行实例与小结	105
2.3.1 MVC 设计模式	34	第 5 章 Struts 的配置	106
2.3.2 Struts 的 MVC 构架	36	5.1 Web.xml 的配置	106
第 3 章 初识 Hibernate	37	5.2 Struts-config.xml 的配置	108
3.1 Hibernate 预备知识	37	5.3 多个配置文件	119
3.1.1 JDBC	37	5.4 理解流程	121

第 6 章 Struts 组件	124	11.1 自动代码生成工具	183
6.1 Struts 的组件	124	11.2 事务与并发	183
6.1.1 Model 组件	124	11.3 延迟加载 (Lazy Loading)	188
6.1.2 View 组件	124	第 12 章 RSS 在线阅读器	190
6.1.3 Controller 组件	125	12.1 RSS 知识	190
6.1.4 组件的工作以及 开发过程	125	12.2 利用 Struts 开发 RSS 在 线阅读器	194
6.2 4 个核心组件	125	12.3 利用 Hibernate 完成 RSS 自动更新	209
6.2.1 Action Servlet	125	12.4 Struts+Hibernate 开发 RSS 阅读器	216
6.2.2 Action Class	126	12.5 总结与改进	224
6.2.3 ActionForm	128	第 13 章 用户管理系统	226
6.2.4 ActionMapping	129	13.1 系统设计	226
第 7 章 Struts 标签	131	13.1.1 功能、流程与模块	226
7.1 标签库	131	13.1.2 数据库设计与 Hibernate 设计	228
7.2 HTML 标签	132	13.1.3 Action 和 Form 设计	228
7.2.1 HTML 标签的使用	133	13.1.4 页面设计	229
7.2.2 主要的 HTML 标签	134	13.2 具体实现	229
7.3 Bean 标签	140	13.2.1 构建系统框架	230
7.4 Logic 标签	142	13.2.2 实现功能模块	237
7.5 Nested 标签	143	13.2.3 运行实例	257
7.6 自定义标签	144	第 14 章 网上论坛	261
第 8 章 Tiles、Validator 和异常	151	14.1 系统设计	261
8.1 Tiles 框架	151	14.1.1 功能、模块与流程	261
8.2 Validator	153	14.1.2 数据库设计与 Hibernate 设计	262
8.3 利用 Struts 插件机制 配置 Hibernate	156	14.1.3 Action 和 Form 设计	263
8.4 Struts 的全局异常与局部异常	158	14.1.4 页面设计	263
8.5 Struts 开发的一些建议	159	14.2 系统实现	264
第 9 章 Hibernate 的映射关联方式	161	14.2.1 构建系统框架	264
9.1 一对一关联	161	14.2.2 实现功能模块	271
9.2 一对多关联	162	14.2.3 实例演示	290
9.3 多对多关联	165	第 15 章 考试成绩维护系统	292
第 10 章 Hibernate 的数据查询和操作	167	15.1 系统设计	292
10.1 HQL	167	15.1.1 功能与模块	292
10.2 条件查询	172	15.1.2 数据库设计 与 Hibernate 设计	294
10.3 SQL 查询	175		
10.4 对持久化对象的操作	175		
第 11 章 Hibernate 自动化生成工具、 事务与并发、延迟加载	183		

15.1.3	Action、Form 和 Manager 类设计	295	16.2	利用 JUnit 做单元测试	322
15.1.4	页面设计	296	16.3	利用 StrutsTestCase 做 Struts 单元测试	326
15.2	系统实现	296	16.4	利用 JMeter 做压力测试	333
15.2.1	构建系统框架	296	16.4.1	压力测试概述	333
15.2.2	实现功能模块	302	16.4.2	JMeter 概述	334
第 16 章	Web 测试	318	16.4.3	应用 JMeter	335
16.1	Web 测试的内容	318	16.5	其他测试工具	340
16.1.1	Web 测试的内容	318	附录 A	HTML 标签描述	341
16.1.2	Web 测试的方法	321	附录 B	Logic 标签描述	378

第 1 章 Web 应用程序与 J2EE 平台

本书主要介绍如何利用 Struts 和 Hibernate 建立基于 J2EE 的 Web 应用程序。J2EE 是 Java 2 平台企业版 (Java 2 Platform, Enterprise Edition) 的简称, 是 SUN 与 IBM 等厂商协同业界共同拟定而成的技术规范, 它是一个以企业与企业之间的运算为导向的 Java 环境。Struts 和 Hibernate 并不包括在 J2EE 标准中, 但它们作为开源的框架, 是在 J2EE 平台上进行开发的一种非常流行的选择, 受到了越来越多的关注。本章首先介绍 J2EE 平台的基础知识。

1.1 Web 应用程序

建立 J2EE 平台的目的是为了开发出企业或者其他客户所需要的 Web 应用程序 (Web application)。Web 应用程序的定义范围很广, 以前最简单的由纯 HTML 静态页面构成的网站也属于其中一种, 但是企业或组织需要的是一种更高效、更能为其创造价值的平台, J2EE 就是为了这个目的而产生的。J2EE 的目标就是建立一个高质量、适合于企业级应用的平台, 当然, 要开发 Web 应用程序, 除了 J2EE 还可以有其他的选择, 比如 Microsoft 的 .Net 框架。

用 J2EE 建立的 Web 应用程序, 其大致结构如图 1-1 所示。

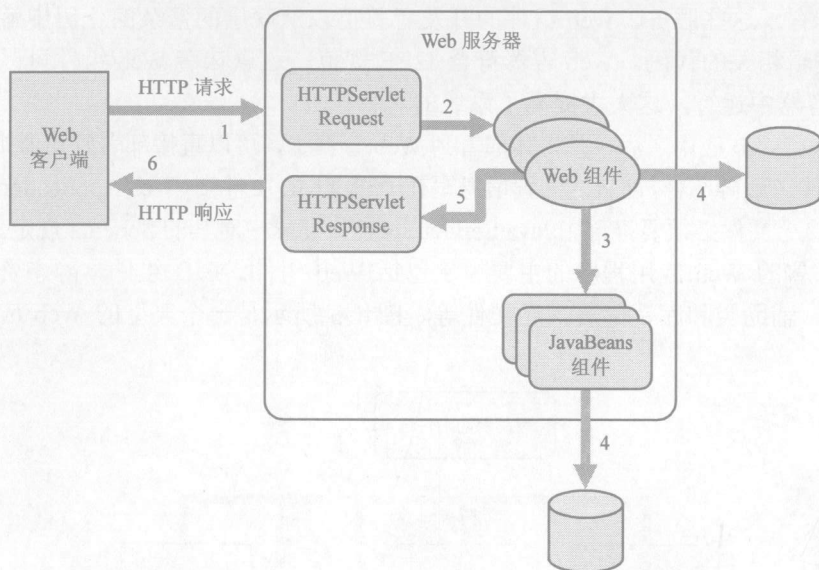


图 1-1

图中的 Web 客户端即浏览器, 比如 IE 或者 Firefox, 它发送 HTTP 请求 (HTTP Request) 给 Web 服务器以请求资源, 然后接收 HTTP 响应并显示响应结果。这就是客户端的行为。

在服务器端, Web 服务器接收到 HTTP 请求之后, 将此请求包装生成相应的 HTTPServletRequest 对象, 并将其传递给 Web 组件 (Web Components) 进行处理, Web 组件

和 `JavaBean` 或数据库等交互，完成所需要的业务逻辑，然后根据结果生成 `HTTPServletResponse` 对象，并由 Web 服务器将 `HTTPServletResponse` 对象转化成 HTTP 响应再传给客户端，或者将请求传递给另外的 Web 组件，由它们最终返回响应。

正是 Web 组件给 Web 应用程序带来了动态性。在 J2EE 平台下，Web 组件是建立在 `Java Servlet` 或 `JSP`（`JavaServer Page`）的基础上的。

`Servlet` 是处理请求和生成响应的 `Java` 类，而 `JSP` 拥有 `Servlet` 的行为（它实际上被转化成 `Servlet`），但对生成显示层提供了更方便的支持。所以在开发过程中，需要面向服务器的，或者用于控制的部分，适合于用 `Servlet`；而在生成表现层的地方，`JSP` 会更合适。

`Servlet` 和 `JSP` 构成 Web 组件的基础，在其之上人们开发了很多框架，这些框架提供了一些通用功能的封装，提高了开发 Web 应用程序的效率。这些框架包括 `JSTL`，`JSF`，以及本书重点讨论的 `Struts`，如图 1-2 所示。

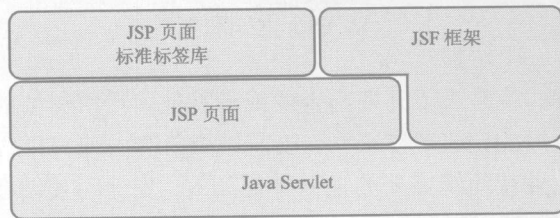


图 1-2

Web 组件的正常工作需要一个运行环境，这就是 Web 容器（`Web Container`）。Web 容器为 Web 组件提供一些底层的服务，这些服务包括：请求传递、安全性、并发控制、生命周期的管理等等。有了这些服务，Web 组件的开发者就可以从底层的繁杂的代码中解脱出来，专心开发业务逻辑相关的代码。Web 容器符合 J2EE 规范，这意味着 Web 组件可以在任何符合规范的 Web 容器中运行，这大大提高了系统的可移植性。

当开发完成 Web 应用之后，要将其部署到 Web 容器中，可以直接部署到容器中，也可以打包成 `WAR` 文件之后再部署。部署的配置信息写在一个 XML 文件中（`Web Application Deployment Descriptor`）。这个文件必须要符合在 `Java Servlet Specification` 中对其的 Schema 规定。

一个可部署的 Web 应用程序的主要模块包括 Web 组件、`EJB` 组件、静态资源文件（比如图像文件）、辅助类和库、部署描述文件等。图 1-3 所示是一个典型的 Web 应用程序的模块组成结构。

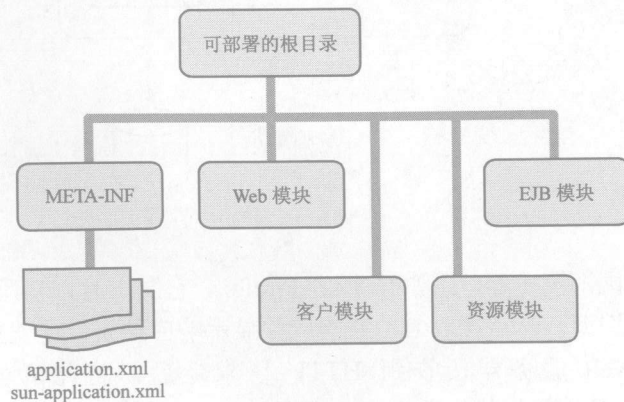


图 1-3

Web 应用程序的开发部署过程如下所示：

- (1) 开发 Web 组件的代码。
- (2) 编写配置信息。
- (3) 编译 Web 组件以及辅助类。
- (4) 将 Web 应用程序打包（可选步骤）。
- (5) 将 Web 应用程序部署到 Web 容器中。

单独可部署的 Web 应用程序，其一般的目录结构如图 1-4 所示。

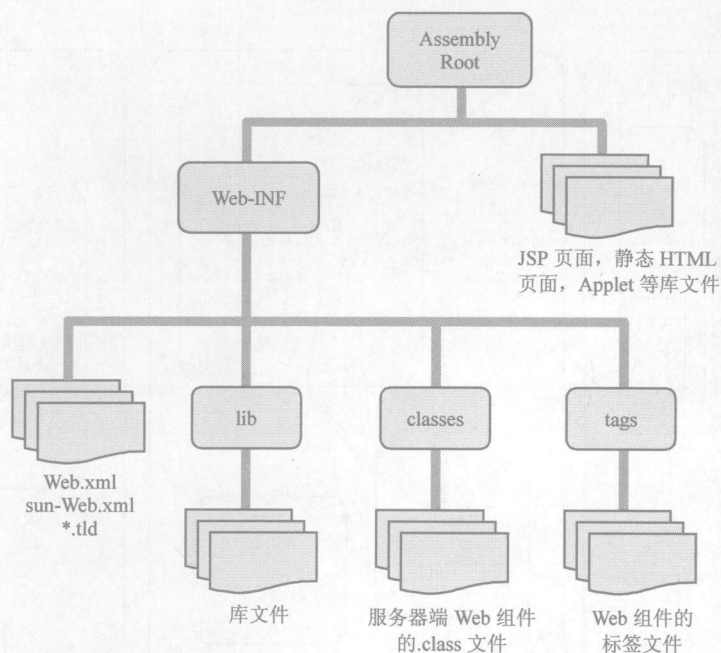


图 1-4

除了 JSP 页面、静态 HTML 页面等，需要特别注意 WEB-INF 目录下的文件，WEB-INF 目录是外部不可见的，其中包含了 Web 应用系统运行时内部所需的文件，包括：

- web.xml，即 Web Application Deployment Descriptor，用以描述和配置此 Web 应用。
- Tags，即标签库。
- Classes，即服务器端的类文件，包括 servlets、utility 类和 JavaBean 等，实现内部逻辑。
- Lib，即 Classes 中所需的 JAR 包。
- 其他配置文件（如 tag 库描述文件等）。

1.2 J2EE 平台的产生

可以说，J2EE 的产生和流行是“聪明人的脑袋和商人的努力相结合的结果”。“聪明人的脑袋”为技术带来了无穷的活力，而“商人的努力”将技术进行整合并在市场上实现它们。

另一角度看，J2EE 的出现是 Internet 从“发布型”向“事务型”转化的必然结果，现在的企业需要的不再是单一的发布信息的平台，它们需要的是能更好地实现商业逻辑的、有更

好的用户体验并且能方便灵活利用后台系统的一个平台，这正是 J2EE 的特点，所以 J2EE 的流行看起来是不可避免的。

J2EE 平台的目标是建立和部署企业级应用程序的统一标准。J2EE 平台出现之后吸引了大量企业级计算机厂商的参与，保证了这个平台能满足最广泛的企业级应用系统的需求。

J2EE 平台利用客户层来提供用户界面，一个或多个中间层组件来提供客户端服务和系统的商业逻辑，利用后台的信息系统层来管理数据。图 1-5 所示是一个典型的 J2EE 系统体系图。

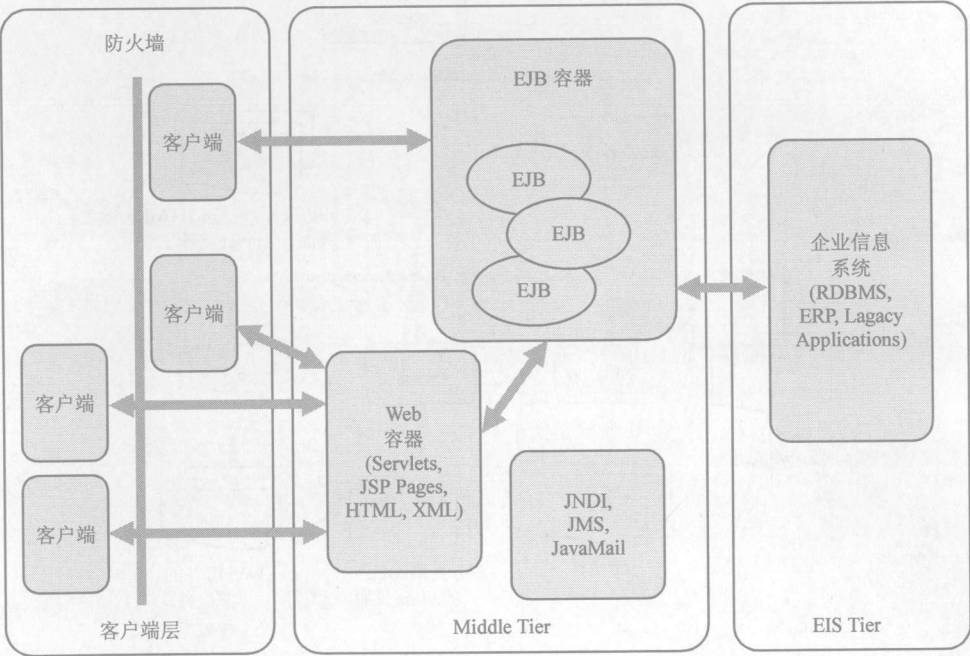


图 1-5

1.3 J2EE 平台特性

1.3.1 作为企业应用的特性

在现代经济中，信息对于企业，已经变得至少跟商品和服务一样重要。信息在企业的长远策略中尤其重要。将信息资本化的能力则是企业成功的关键。对于企业来说，信息管理一直是一个需要思考的问题。在计算机化的管理中，需要考虑很多问题——集成性、信息发掘、企业内部以及与外部企业的信息传输、安全性等。现在的 J2EE 平台对于这些问题提供了有效的、灵活的、分布式的解决方案。

J2EE 平台的目标是定义一个标准，实现这个标准可以帮助企业在信息经济市场竞争中具备更强的竞争力。J2EE 平台是采用了一系列不断改进的新技术的一种分布式企业应用系统，它通过采用基于组件的开发模型来加快开发速度。J2EE 模型支持从传统的、合作企业内部的

应用程序，到在互联网上的电子商务网站的一系列系统。

采用 J2EE 平台带来的优势包括如下内容。

(1) 加快了开发速度。

客户从提出需求到得到响应的时间是很重要的因素。企业需要更快地开发和部署他们的企业应用程序，并根据用户的需求方便快速地对系统进行修改、完善和升级。所以这个系统需要简单而又有效率地把原来已有的系统集成起来，并且要具有良好的可扩展性，以迎接将来不断变化的市场需求。J2EE 的分层体系结构就能很好地满足这个要求。

同时企业级应用程序需要很快地从原型发展到产品，并且快速地在产品的生命周期中不断完善。而“编写一次，随处运行”的特性，使得 J2EE 系统能更方便地测试与部署，从而大大提高了开发效率。

(2) 可扩展性。

企业级的应用程序需要有很强的扩展性，企业会面临需要将它们的 Web 应用程序扩大规模的情况，这个时候，如果采用了不合理的技术，就会造成很大的成本和技术风险。而 J2EE 平台标准保证了企业依据此标准建立的系统能比较方便地进行扩展，J2EE 平台采用了多层结构，在各层之间提供了方便的资源管理和服务，比如数据库的连接等，这样开发人员就不用顾忌数据库连接的负载平衡等问题了。同时，可以根据客户端的类型和数量级来确定所用的服务器，并且在需要的时候变更服务器而不需要对系统本身作出太大的改动。

(3) 组件模型带来简化的构架。

J2EE 平台可以在任何符合标准的服务器上运行，并且基于组件的 J2EE 开发模型能更容易将需求确定成功能，并且由于只需要升级必要的组件，所以系统的升级比较方便。

组件能在运行的时候根据配置来连接其他的组件，有了可配置的组件行为，开发人员就不需要重写代码，开发人员通过这些配置来与组件所在的服务器交流，这个过程也可以通过自动化的工具来实现。

(4) 与现有系统集成。

企业的数据是多年的积累和投资，往往位于一个比较古老的系统中，所以企业级应用系统的开发人员会面临一个重大的考验，就是如何利用并集成那些多年来一直使用的数据系统。为了实现这个目标，J2EE 平台通过中间层和后台服务来访问这些现有系统。具体来说，J2EE 平台提供了下面这些技术来与现有的信息系统集成。

- JDBC 是用 Java 访问关系数据库的 API。
- The Java Transaction API (JTA) 是在各个异种的企业信息系统间管理和协调事务的 API。
- The Java Naming and Directory Interface (JNDI) 是访问现有信息系统的命名和目录服务 API。
- The Java Message Service (JMS) 是利用 IBM MQ 或 TIBCO Rendezvous 这样的系统来发送和接收消息的 API。
- Java IDL 是调用 CORBA 的 API。

(5) 服务器、工具和组件的自由选择。

开发企业应用程序，需要根据手头的项目以及技术能力和投资成本，选择并配置应用系

统，以得到最优化的效果。J2EE 平台为此提供了多种自由的选择，这些选择包括服务器、组件和其他工具等。

(6) 更专业化的开发人员。

J2EE 的基于组件的开发模式能够根据不同的技能来细化开发人员的职能。所有参与开发过程的人员能更好地发挥他们的特长，JSP 模板的设计者能专心于他的工作，而商业逻辑的开发者、部署人员也能专注于自己擅长的领域。这个特点对于程序的升级也有好处，一个 J2EE 系统最常改变的一般是前台的用户界面，这样网页设计师就可以着力于页面的设计而不需要编程能力。当然这些职能的划分在不同的开发小组中会有不同，在有些项目中，或许一两个人就能把所有的职能角色都分配掉了。

1.3.2 J2EE 的多层结构

J2EE 平台提供了一种多层的分布式模型，这意味着企业级应用系统的各个部分可以运行在不同的设备上。J2EE 框架定义了客户端层、中间层（可以由一个或多个子层组成）和对现有数据系统提供支持服务的后端层。客户端层提供了对很多种类的客户端的支持，包括企业防火墙内部和外部的客户端。中间层通过在 Web 容器中的 Web 层提供与客户端的交互，并通过 EJB 等进行商业逻辑。后台的企业信息层（The Enterprise Information System, EIS）提供各种 API 以对现有数据系统提供服务。

图 1-6 显示了 3 个层次的主要组件。

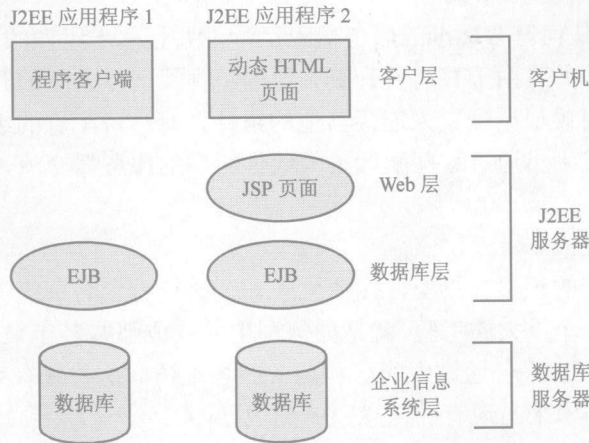


图 1-6

J2EE 客户端层支持企业防火墙内部和外部各种类型的客户端。客户端可以是浏览器（以得到纯 HTML 文件、由 JSP 或 Servlet 动态生成的 HTML 页面）或是 Java Applet，也可以是独立的 Java 程序。当然，当大家提到 J2EE 客户端时首先指的还是 Web 方式的。

为了提供比较复杂的用户交互，在客户端层需要直接提供一些功能性的服务，这通常是利用 Java Bean 组件来实现向中间层请求服务的。这些提供服务的客户端 Java Bean 通常以 Java Applet 形式自动下载到用户的浏览器。这些客户端的 bean 可以看作是用 java 写的独立程序，它们的任务就是提供给客户端必要的服务。针对不同的平台，这些 bean 的提供者需要编写不同的安装文件并让它们能够自动下载到用户所在的平台，虽然这些 java bean 本身是和平

台无关的，但是安装文件的代码一般都是和操作系统相关的。

实际上，客户端也完全可以采用 Java 语言之外的其他语言编写，比如用 Visual Basic 写的程序同样可以访问用 servlet 实现的中间层——由于 servlet 接受标准的 HTTP 请求，所以可以用任何平台上的任何语言编写的程序来得到中间层的服务。

在多层结构中（或一般三层结构），中间层扮演了关键的角色。它接收客户端的请求，把客户端从复杂的与数据库等后台的交互中解脱出来，这样客户端就不再需要作数据库查询，执行复杂的商业逻辑或连接到古老的 legacy 系统——中间层会替它们完成这些工作。

中间层的任务是执行商业逻辑，比如一个信用卡网站，所有业务相关的操作都由中间层完成。同时中间层也需要提供一些系统级的服务，这些服务包括：

- 提供客户端到后台系统的远程访问能力。
- 线程和事务管理。
- 安全性。
- 资源池。

正是由于中间层提供了这些服务，才为瘦客户端的应用提供了可能。对用户来说他们能更轻巧、快速和简单地访问 Web 应用程序，而企业也避免了改造原有后台系统的成本。

图 1-7 所示是三层结构之间交互的示意图。

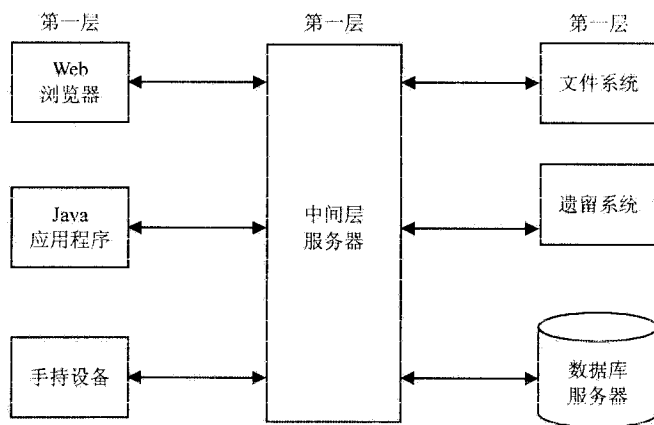


图 1-7

1.3.3 J2EE 的结构变形

用 J2EE 平台开发的 Web 应用程序，根据具体的需求和规模，可以采用不同的体系结构，主要有以下几种类型。

(1) 由浏览器和应用程序共同组成客户端，由 Web 容器和 EJB 容器组成中间层，如图 1-8 所示。

(2) 由浏览器组成客户端，由 Web 容器和 EJB 容器组成中间层，如图 1-9 所示。

(3) 由客户端的应用程序直接和 Web 组件、EJB 或后台数据系统交互，如图 1-10 所示。

(4) 客户端直接与 EJB 交互，如图 1-11 所示。

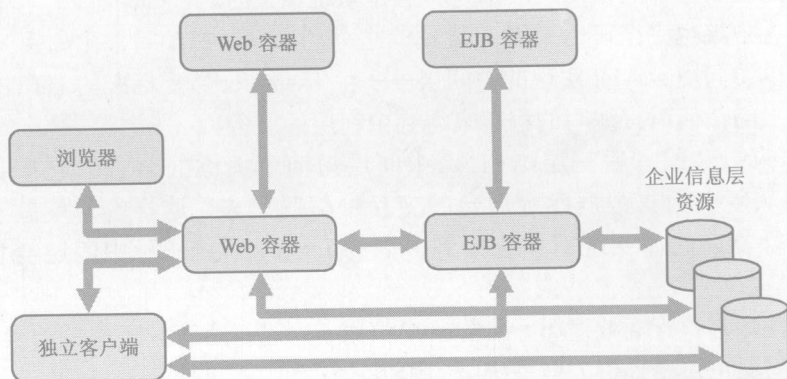


图 1-8

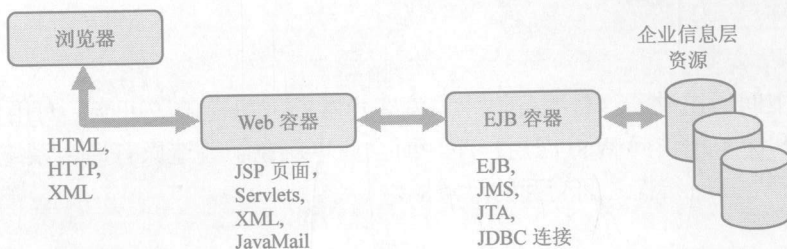


图 1-9

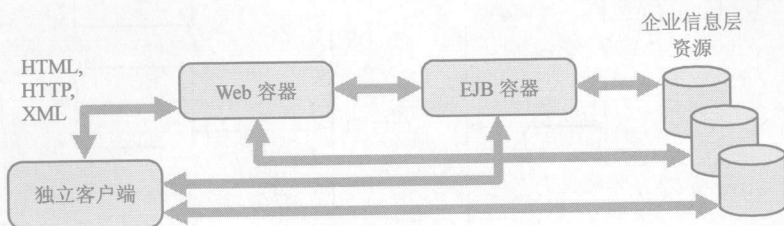


图 1-10

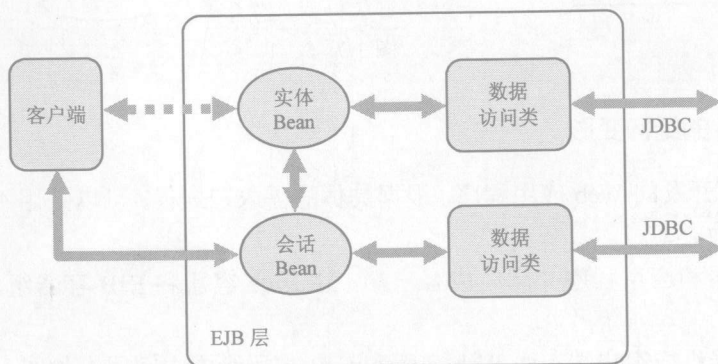


图 1-11

(5) 仅有 Web 容器，如图 1-12 所示。

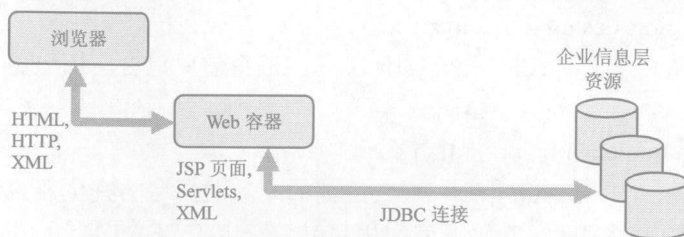


图 1-12

(6) 采用其他数据访问方式的结构, 如图 1-13 所示。

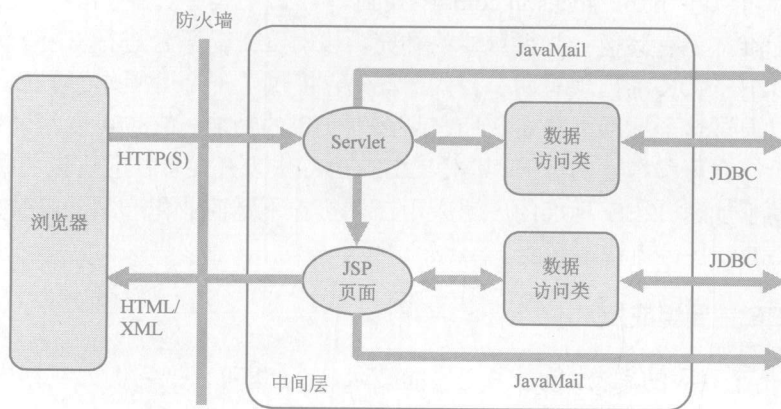


图 1-13

(7) 没有纯粹的客户端的结构, 如图 1-14 所示。

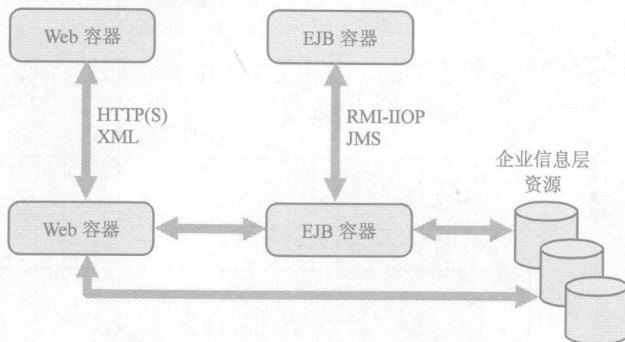


图 1-14

可以看出, J2EE 灵活的结构变形使得它具备了很强的灵活性。本书要讨论的 Struts+Hibernate 的结构就是其中非常流行的一种, Struts 在 JSP 和 Servlet 之上提供了更方便的 Web 组件开发模式, 而 Hibernate 作为数据持久化的方法也完全能够取代 EJB, 并且有着更好的性能。

1.4 J2EE 平台简介

J2EE 平台由 4 个关键部分定义: Specification、Reference implementation、Compatibility test