

软件工程系列规划教材

孙 明 詹瑾瑜 陈 波 / 编著



Java Web

开发技术

Java Web Kaifa Jishu



电子科技大学出版社

软件工程系列规划教材

Java Web

开发技术

Java Web Kaifa Jishu

孙 明 詹瑾瑜 陈 波 / 编著

ISBN 7-302-11002-1

出 版 社
编 审 组
主 编
副 主 编
参 审 员
审 定 人
审 定 组
审 定 人
审 定 组
审 定 人
审 定 组



电子科技大学出版社

图书在版编目(CIP)数据

Java Web 开发技术 / 孙明, 詹瑾瑜, 陈波编著.

—成都: 电子科技大学出版社, 2012. 4

ISBN 978-7-5647-1135-1

I. ①J… II. ①孙… ②詹… ③陈… III. ①

JAVA 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2012) 第 066127 号

内容提要

本书详细介绍了 Java Web 开发中必不可少的基础知识, 主要讲解包括 Java Web 开发中常用的各种框架和新兴技术, 包括 Struts 2.0、Spring、Hibernate 和 Ajax 等, 这些技术在实际的项目开发中应用广泛。最后本书结合 Struts 2.0+Spring+Hibernate 开发了一个虚拟文件管理系统, 在系统中展示了 Struts 2.0、Spring、Hibernate 的具体使用方法。

本书是在吸取了国内外有关教材的精华, 并结合编者多年教学的基础上编写的, 内容注重科学性、先进性, 强调实用性。本书语言简练、内容实用, 各个章节中都提供了丰富的示例程序, 可以让读者快速掌握 Java Web 开发的各种技巧。

本书是高等院校及各类培训学校的 Java Web 开发课程教材, 也可作为 Java Web 开发人员教学参考书和技术手册。

Java Web 开发技术

孙 明 詹瑾瑜 陈 波 编著

出 版: 电子科技大学出版社 (成都市一环路东一段 159 号电子信息产业大厦 邮编: 610051)

策划编辑: 杨仪玮

责任编辑: 岳 慧

主 页: www.uestcp.com.cn

电子邮箱: uestcp@uestcp.com.cn

发 行: 新华书店经销

印 刷: 四川经纬印务有限公司

成品尺寸: 185 mm×260 mm 印张 19.5 字数 475 千字

版 次: 2012 年 4 月第一版

印 次: 2012 年 4 月第一次印刷

书 号: ISBN 978-7-5647-1135-1

定 价: 45.00 元

■ 版权所有 侵权必究 ■

◆ 邮购本书请与本社发行部联系。电话: (028) 83202323, 83256027。

◆ 本书如有缺页、破损、装订错误, 请寄回印刷厂调换。

前言

随着互联网的飞速发展，软件开发模式也由传统的 C/S（Clients/Server，客户/服务器）模式转化为现在的 B/S（Browser/Server，浏览器/服务器）模式。作为一种安全性最好的语言，Java 成为服务器端的最好选择，于是在各种 Web 应用程序中，纷纷采用 Java 作为首选的编程语言。

在 Java Web 开发的领域中，不仅有 Oracle 公司提供强大的技术支持，而且还有大量的开源社区推出各种开源的 Java 项目，这一切都使得 Java Web 的开发日益繁荣。而精通 Java Web 开发的软件工程师也越来越受各大软件公司的欢迎。

为了帮助读者快速掌握 Java Web 开发这项技术，编者精心编著了本书。本书依照读者的学习规律，首先介绍基本概念和基础知识，在读者掌握了这些基本概念和基础知识的基础上，再提供简单的示例，使读者对知识点有整体上的认识，然后再对内容进行深入的讲解，严格遵循由浅入深、循序渐进的原则。

本书的主要内容是讲解 Java Web 开发中常用的各种技术，这些技术都是在实际的项目开发中必不可少的，通过本书的内容，读者可以快速掌握 Java Web 开发的基本技术，为今后就业打好基础。本书由浅入深、循序渐进地介绍了 Java Web 开发中的各种实用技术。读者仅仅需要具备基本的 Java 语法知识就可以很快掌握本书的内容。本书特点如下：

1. 由浅入深、循序渐进

本书中的讲解遵循由浅入深、循序渐进的原则。不仅在总体的章节划分上把整书分为基础篇、进阶篇和实战篇，由浅入深地讲解 Java Web 开发的知识，而且在每一个章节、每一个知识点的讲解中都是从最基础的内容开始，然后逐步深入到核心的内容，使读者比较容易接受这些新的知识。

2. 实例经典、丰富实用

本书的各个章节都提供了丰富的示例程序，这是本书另一个特色。本书减少了纯理论介绍的比重，增大了示例程序的比重，强调通过示例程序展示具体的知识点。读者可以通过这些示例程序体会到知识点的具体实用方法，这样的效果要比纯文字的描述更容易理解。本书的示例程序都是编者调试通过的，读者可以在计算机上直接运行。

3. 讲解详细、内容充实

本书不仅详细讲解了 Java Web 开发的各種基础知识，还介绍了在 Java Web 开发中常用的各种实用技术，通过具体的示例项目展示了这些技术的使用方法。读者通过本书可以了解目前 Java Web 开发中所用到的各种知识。

本书包括以下主要内容：

第一章介绍了 Java EE 开发的背景知识，包括 Java EE 的模型、结构和核心 API 等基本知识。

第二章介绍了 Java EE 开发环境的配置,包括 JDK、Eclipse、MyEclipse、Tomcat 等的安装,最后介绍了在开发中常用的调试方式 JUnit。

第三章介绍了 JSP 技术,包括 JSP 的语法、生命周期、基本的内置组件等内容。

第四章介绍了 Servlet 编程,包括 Servlet 的工作原理、基本结构、生命周期等,最后本章详细介绍了 Servlet 在实际中的应用。

第五章介绍了 JDBC 数据库编程,包括数据库开发环境搭建、SQL 基础知识、JDBC 编程及数据库连接池。

第六章介绍了 Java EE 的 Struts 2.0 框架,包括 Struts 2.0 的配置、拦截器、标签库、输入校验、上传下载文件及 Struts 2.0 国际化的一些知识。

第七章介绍了 Java EE 的 Spring 框架,包括 Spring 的配置、控制反转、及切面编程。

第八章介绍了 Java EE 的 Hibernate 框架,包括 Hibernate 的配置、持久化数据、检索方式及一些开发模式。

第九章介绍了 Ajax 的技术,包括 Ajax 的工作原理、XMLHttpRequest 对象、客户端与服务器端请求响应等知识。

第十章综合使用 Struts 2.0、Spring、Hibernate 构建 mdvsf 系统,对系统的设计到完成给出了详细方法和步骤。

本书是高等院校及各类培训学校的 Java Web 开发课程教材,也可作为 Java Web 开发人员教学参考书和技术手册。

本书由孙明、詹瑾瑜、陈波等共同编写,其中第 1 章、第 6 章、第 7 章、第 8 章由孙明编写,第 2 章、第 5 章、第 10 章由詹瑾瑜编写,第 3 章、第 4 章、第 9 章由陈波编写。本书由孙明主编及审核。

编 者
2012-3-23

目 录

第 1 章	Java EE 基础	1
1.1	Java EE 简介	1
1.2	Java EE 的优势	1
1.3	Java EE 的四层模型	3
1.4	Java EE 的结构	4
1.5	Java EE 的核心 API 与组件	5
1.6	Java EE 相关名词	6
1.7	本章小结	7
	习题	8
第 2 章	配置软件开发环境	9
2.1	安装 JDK	9
2.1.1	在 Windows 系统中安装 JDK	12
2.1.2	配置 JDK 的参数	12
2.1.3	在控制台编写程序	14
2.2	安装 Eclipse 和 MyEclipse	14
2.2.1	Eclipse 的简介和安装	14
2.2.2	MyEclipse 的简介和安装	17
2.3	安装 Tomcat	18
2.3.1	获得 Tomcat	19
2.3.2	安装 Tomcat	19
2.4	JUnit	20
2.4.1	JUnit 特性	21
2.4.2	JUnit 优点	21
2.4.3	安装 JUnit	21
2.4.4	如何使用 JUnit 写测试	22
2.4.5	JUnit 最佳实践	23
2.5	本章小结	25
	习题	25
第 3 章	JSP 技术	26
3.1	JSP 简介	26

3.1.1	什么是 JSP	26
3.1.2	JSP 的优点	26
3.2	JSP 语法	28
3.2.1	JSP 页面中元素	28
3.2.2	JSP 指令	28
3.2.3	JSP 声明	32
3.2.4	Java 程序片段 (Scriptlet)	32
3.2.5	JSP 表达式	34
3.2.6	JSP 中的注释	35
3.3	JSP 生命周期	35
3.4	JSP 九种基本内置组件	37
3.4.1	request 对象	38
3.4.2	response 对象	42
3.4.3	session 对象	44
3.4.4	application 对象	46
3.4.5	out 对象	47
3.4.6	pageContext 对象	48
3.4.7	config 对象	49
3.4.8	page 对象	51
3.4.9	exception 对象	51
3.5	本章小节	53
	习题	53
第 4 章	Servlet 编程	54
4.1	Servlet 简介 (Server + applet = Servlet)	54
4.1.1	Servlet 工作原理	54
4.1.2	Servlet 基本结构	55
4.1.3	Servlet 的生命周期	56
4.1.4	运行你的第一个 Servlet 实例	57
4.2	Servlet 实际应用	60
4.2.1	Servlet 与 JSP 的关系	60
4.2.2	使用 Servlet 实现 MVC 开发模式	61
4.2.3	JSP 页面调用 Servlet 的方法	62
4.2.4	Servlet 中的文件操作	67
4.2.5	Servlet 获取请求信息头部内容	71
4.3	本章小结	73
	习题	73

第 5 章 JDBC 数据库编程	74
5.1 JDBC	74
5.2 数据库开发环境搭建	75
5.2.1 MySQL 数据库的下载和安装	75
5.2.2 MySQL 数据库安装测试	80
5.3 SQL 基础知识	81
5.3.1 SQL 的特点	81
5.3.2 简述 SQL 的六种基本指令	82
5.3.3 创建和删除数据库	82
5.3.4 创建、删除、修改数据库表	83
5.3.5 约束概述	87
5.3.6 数据操纵语言 (DML)	87
5.3.7 数据查询语言	90
5.4 JDBC 编程	94
5.4.1 JDBC 编程步骤	94
5.4.2 JDBC 编程实例	95
5.5 更多 JDBC 编程	97
5.5.1 预处理	97
5.5.2 批处理命令	99
5.5.3 事务处理	100
5.6 数据库连接池	101
5.6.1 数据库开发中面临的数据库资源的问题	101
5.6.2 数据库连接池工作原理	102
5.6.3 常用数据库连接池介绍	103
5.7 本章小结	104
习题	104
第 6 章 Struts 2.0	105
6.1 Struts 2.0 简介	105
6.1.1 Struts 2.0 的下载与安装	105
6.1.2 Struts 2.0 项目组成	105
6.2 Struts 2.0 配置	106
6.2.1 web.xml 的配置	106
6.2.2 配置文件 struts.xml 和 struts.properties	107
6.2.3 Struts 2.0 中的拦截器	114
6.3 Struts 2.0 中的标签库	121
6.3.1 Struts 标签库简介	121
6.3.2 使用控制标签	122

6.3.3	使用表单 UI 标签	122
6.3.4	Bean 标签库	124
6.4	输入 校验	124
6.4.1	Struts 2.0 校验机制	125
6.4.2	利用 Struts 2.0 框架实现校验	125
6.4.3	内置校验器	128
6.4.4	校验器配置	132
6.4.5	校验器的执行与短路	135
6.4.6	自定义校验器	135
6.5	上传和下载文件	136
6.5.1	文件上传的原理	136
6.5.2	Struts 2.0 的文件上传	139
6.5.3	同时上传多个文件	140
6.5.4	使用 Struts 2.0 的控制文件下载	143
6.6	Struts 2.0 的国际化	144
6.7	本章小结	147
	习题	147
第 7 章	Spring	148
7.1	Spring 简介	148
7.1.1	Spring 模块介绍	148
7.1.2	Spring 安装配置	150
7.1.3	Spring 的优点	153
7.2	控制反转 (IOC)	154
7.2.1	IOC 容器简介	154
7.2.2	IOC 示例入门	154
7.2.3	依赖注入的几种形式	156
7.2.4	集合类型数据注入	163
7.2.5	延迟初始化 Bean	165
7.2.6	实例化 Bean	165
7.3	面向切面编程(AOP)	168
7.3.1	AOP 概述	168
7.3.2	AOP 重要概念	169
7.3.3	切面的声明	170
7.3.4	AOP 的优势	172
7.3.5	AspectJ 具体实现 AOP	173
7.4	本章小节	176
	习题	177

第 8 章 Hibernate.....	178
8.1 Hibernate 简介.....	178
8.1.1 理解对象持久化.....	178
8.1.2 使用 JDBC.....	178
8.1.3 Hibernate 环境与配置.....	180
8.2 Hibernate 基础.....	180
8.2.1 简单入门.....	180
8.2.2 持久化对象.....	184
8.2.3 检索对象.....	185
8.2.4 高级配置.....	186
8.2.5 继承.....	188
8.3 操作持久化数据.....	188
8.3.1 构建 SessionFactory.....	188
8.3.2 DAO 模式.....	190
8.3.3 利用 DAO 来操作数据.....	191
8.3.4 DAO 模式的实战应用.....	194
8.4 Hibernate 检索方式.....	196
8.4.1 HQL 查询.....	196
8.4.2 QBC 查询.....	202
8.4.3 SQL 查询.....	207
8.5 本章小结.....	209
习题.....	209
第 9 章 Ajax.....	210
9.1 Ajax 技术简介.....	210
9.2 Ajax 工作原理.....	211
9.3 XMLHttpRequest 对象.....	213
9.3.1 XMLHttpRequest 对象属性.....	213
9.3.3 XMLHttpRequest 方法及其属性.....	214
9.4 客户端向服务器发送请求.....	215
9.5 服务器端响应客户端请求.....	217
9.5.1 在服务器端响应用户请求.....	217
9.5.2 返回 XML 格式的响应文档.....	218
9.6 客户端处理服务器响应.....	223
9.6.1 JavaScript 的 DOM 绑定.....	223
9.6.2 利用 DOM 进行 Web 响应.....	234
9.7 本章小节.....	235
习题.....	235

第 10 章 虚拟数据文件管理系统	237
10.1 系统概述及架构设计	237
10.2 持久层设计	241
10.2.1 系统实体	241
10.2.2 系统 E-R 图与数据表	242
10.2.3 实现 Hibernate PO	245
10.3 实现系统 DAO 层	251
10.3.1 DAO 接口	252
10.3.2 DAO 实现	253
10.3.3 配置系统 DAO 组件	264
10.4 实现业务逻辑层	265
10.5 实现系统 Web 层	268
10.5.1 Struts 2.0 的配置	268
10.5.2 用户操作部分	277
10.5.3 管理员操作部分	290
10.6 本章小结	298
习题	298
参考文献	299

第1章 Java EE 基础

Java EE 是一套全然不同于传统应用开发的技术架构，其包含许多组件，主要可简化且规范应用系统的开发与部署，进而提高可移植性、安全与再用价值。

1.1 Java EE 简介

1. Java EE Java2 平台企业版 (Java 2 Platform, Enterprise Edition)

Java EE 核心是一组技术规范与指南，其中所包含的各类组件、服务架构及技术层次，均有共同的标准及规格，让各种依循 Java EE 架构的不同平台之间，存在良好的兼容性，解决过去企业后端使用的信息产品彼此之间无法兼容、企业内部或外部难以互通的窘境。

2. Java EE 的概念

目前，Java 2 平台有 3 个版本，分别是适用于小型设备和智能卡的 Java 2 平台 Micro 版 (Java 2 Platform Micro Edition, J2ME)、适用于桌面系统的 Java 2 平台标准版 (Java 2 Platform Standard Edition, J2SE)、适用于创建服务器应用程序和服务的 Java 2 平台企业版 (Java 2 Platform Enterprise Edition, Java EE)。Java EE 是一种利用 Java 2 平台来简化企业解决方案的开发、部署和管理相关的复杂问题的体系结构。Java EE 技术的基础就是核心 Java 平台或 Java 2 平台的标准版。Java EE 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，等等，同时还提供了对 EJB (Enterprise JavaBeans)、Java Servlets API、JSP (Java Server Pages) 以及 XML 技术的全面支持，其最终目的就是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。

Java EE 体系结构提供中间层集成框架，用来满足无需太多费用而又需要高可用性、高可靠性以及可扩展性的应用的需求。通过提供统一的开发平台，Java EE 降低了开发多层应用的费用和复杂性，同时对现有应用程序集成提供强有力支持，完全支持 Enterprise JavaBeans，有良好的向导支持打包和部署应用，可添加目录支持，增强了安全机制，提高了性能。

1.2 Java EE 的优势

Java EE 为搭建具有可伸缩性、灵活性、易维护性的商务系统提供了良好的机制。

1. 保留现存的 IT 资产

由于企业必须适应新的商业需求, 因此利用已有的企业信息系统方面的投资比重新制订全盘方案更为重要。因此, 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。Java EE 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 以及 Netscape Application Server, 这是因为 Java EE 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必废弃已有投资, 即可进入可移植的 Java EE 领域的升级途径。由于基于 Java EE 平台的产品几乎能够在任何操作系统和硬件配置上运行, 因此现有的操作系统和硬件也能被保留使用。

2. 高效的开发

Java EE 允许公司把一些通用的、很繁琐的服务端任务交给中间供应商去完成, 这样开发人员可以集中精力在如何创建商业逻辑上, 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: 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; 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, 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; 分布式共享数据对象 CACHE 服务——让开发人员编制高性能的系统, 极大提高整体部署的伸缩性。

3. 支持异构环境

Java EE 能够开发部署在异构环境中的可移植程序。基于 Java EE 的应用程序不依赖任何特定操作系统、中间件、硬件, 因此设计合理的基于 Java EE 的程序只需开发一次就可部署到各种平台。这在典型的异构企业计算环境中是十分关键的。Java EE 标准也允许客户订购与 Java EE 兼容的第三方的现成的组件, 把它们部署到异构环境中, 节省了由自己制订整个方案所需的费用。

4. 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系统上进行商业运作的大批新客户。基于 Java EE 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(这是 NT 服务器所望尘莫及的)。Java EE 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。

5. 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, 便会有灾难性后果。Java EE 支持长期的可用性, 可将其部署到可靠的操作环境中。例如一些 Java EE 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。最健壮的操作系统的可用性可达到 99.999% 的可用性或每年只需 5 分钟停机时间。这是实时性很强的商业系统理想的选择。

1.3 Java EE 的四层模型

Java EE 使用多层的分布式应用模型,应用逻辑按功能划分为组件,各个应用组件根据它们所在的层分布在不同的机器上。事实上,Sun 设计 Java EE 的初衷正是为了解决两层模式(Client/Server)的弊端,在传统模式中,客户端担当了过多的角色而显得臃肿,在这种模式中,第一次部署的时候比较容易,但难于升级或改进,可伸展性也不理想,而且经常基于某种专有的协议——通常是某种数据库协议。它使得重用业务逻辑和界面逻辑非常困难。现在 Java EE 的多层企业级应用模型将两层化模型中的不同层面切分成许多层。一个多层次应用能够为不同的每种服务提供一个独立的层,以下是 Java EE 典型的四层结构:

- 运行在客户端机器上的客户层组件
- 运行在 Java EE 服务器上的 Web 层组件
- 运行在 Java EE 服务器上的业务逻辑层组件
- 运行在 EIS 服务器上的企业信息系统(Enterprise Information System)层软件

Java EE 应用程序是由组件构成的。Java EE 组件是具有独立功能的软件单元,它们通过相关的类和文件组装成 Java EE 应用程序,并与其他组件交互。Java EE 说明书中定义了以下的 Java EE 组件:

- 应用客户端程序和 Applets 是客户层组件。
- Java Servlet 和 Java Server Pages (JSP) 是 Web 层组件。
- Enterprise Java Beans (EJB) 是业务层组件。

Java EE 的四层模型为:

1. 客户层组件

Java EE 应用程序可以是基于 Web 方式的,也可以是基于传统方式的。

2. Web 层组件

Java EE Web 层组件可以是 JSP 页面或 Servlets。按照 Java EE 规范,静态的 HTML 页面和 Applets 不算是 Web 层组件。

Web 层可能包含某些 Java Bean 对象来处理用户输入,并把输入发送给运行在业务层上的 Enterprise Bean 来进行处理。

3. 业务层组件

业务层代码的逻辑用来满足银行、零售、金融等特殊商务领域的需要,由运行在业务层上的 Enterprise Bean 进行处理。

有三种企业级的 Bean: 会话(session) Bean, 实体(entity) Bean 和消息驱动(message-driven) Bean。会话 Bean 表示与客户端程序的临时交互。当客户端程序执行完后,会话 Bean 和相关数据就会消失。相反,实体 Bean 表示数据库的表中一行永久的记录。当客户端程序中止或服务器关闭时,就会有潜在的服务保证实体 Bean 的数据得以保存。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。

4. 企业信息系统层

企业信息系统层处理企业信息系统软件包括企业基础建设系统例如企业资源计划(ERP), 大型机事务处理, 数据库系统, 和其他的遗留信息系统。例如, Java EE 应用组件可能为了数据库连接需要访问企业信息系统。

1.4 Java EE 的结构

这种基于组件, 具有平台无关性的 Java EE 结构使得 Java EE 程序的编写十分简单, 业务逻辑被封装成可复用的组件, 并且 Java EE 服务器以容器的形式为所有的组件类型提供后台服务。

容器和服务容器设置定制了 Java EE 服务器所提供得内在支持, 包括安全、事务管理、JNDI (Java Naming and Directory Interface) 寻址、远程连接等服务, 以下列出最重要的几种服务:

Java EE 安全 (Security) 模型可配置 Web 组件或 Enterprise Bean, 这样只有被授权的用户才能访问系统资源。每一客户属于一个特别的角色, 而每个角色只允许特定的激活方法。在 Enterprise Bean 的布置描述中应声明角色和可被激活的方法。而不必编写加强安全性的规则。

Java EE 事务管理 (Transaction Management) 模型让你指定组成一个事务中所有方法间的关系, 这样一个事务中的所有方法被当成一个单一的单元。当客户端激活一个 Enterprise Bean 中的方法, 容器介入一管理事务。因有容器管理事务, 在 Enterprise Bean 中不必对事务的边界进行编码。要求控制分布式事务的代码会非常复杂。你只需在布置描述文件中声明 Enterprise Bean 的事务属性, 而不用编写并调试复杂的代码。容器将读此文件并为你处理此 Enterprise Bean 的事务。JNDI 寻址(JNDI Lookup)服务向企业内的多重名字和目录服务提供了一个统一的接口, 这样应用程序组件可以访问名字和目录服务。

Java EE 远程连接 (Remote Client Connectivity) 模型管理客户端和 Enterprise Bean 间的低层交互。当一个 Enterprise Bean 创建后, 客户端可以调用它的方法就像它和客户端位于同一虚拟机上一样。

生存周期管理 (Life Cycle Management) 模型管理 Enterprise Bean 的创建和移除, 一个 Enterprise Bean 在其生存周期中将会历经几种状态。容器创建 Enterprise Bean 后, 若要删除 Enterprise Bean, 首先是在可用实例池与活动状态中移除它, 然后再从容器中移除它, 这是因为调用 Enterprise Bean 的 Create 及 remove 方法时, 容器会在后台执行这些任务。

数据库连接池 (Database Connection Pooling) 模型是一个有价值的资源。获取数据库连接是一项耗时的工作, 而且连接数非常有限。容器通过管理连接池来缓和这些问题。Enterprise Bean 可从池中迅速获取连接。在 Bean 释放连接之可为其他 Bean 使用。

Java EE 应用组件可以安装部署到以下几种容器中去:

- EJB 容器管理所有 Java EE 应用程序中企业级 Bean 的执行。Enterprise Bean 和它们的容器运行在 Java EE 服务器上。
- Web 容器管理所有 Java EE 应用程序中 JSP 页面和 Servlet 组件的执行。Web 组

件和它们的容器运行在 Java EE 服务器上。应用程序客户端容器管理所有 Java EE 应用程序中应用程序客户端组件的执行。应用程序客户端和它们的容器运行在 Java EE 服务器上。Applet 容器是运行在客户端机器上的 Web 浏览器和 Java 插件的结合。

1.5 Java EE 的核心 API 与组件

Java EE 平台由一整套服务 (Services)、应用程序接口 (APIs) 和协议构成, 它对开发基于 Web 的多层应用提供了功能支持, 下面对 Java EE 中的 13 种技术规范进行简单的描述 (限于篇幅, 这里只能进行简单的描述):

1. JDBC (Java Database Connectivity)

JDBC API 为访问不同的数据库提供了一种统一的途径, 像 ODBC 一样, JDBC 对开发者屏蔽了一些细节问题, 另外, JDBC 对数据库的访问也具有平台无关性。

2. JNDI (Java Naming and Directory Interface)

JNDI API 被用于执行名字和目录服务。它提供了一致的模型来存取和操作企业级的资源如 DNS 和 LDAP, 本地文件系统, 或应用服务器中的对象。

3. EJB (Enterprise Java Beans)

Java EE 技术之所以赢得媒体广泛重视的原因之一就是 EJB。它们提供了一个框架来开发和实施分布式商务逻辑, 由此很显著地简化了具有可伸缩性和高度复杂的企业级应用的开发。EJB 规范定义了 EJB 组件在何时如何与它们的容器进行交互作用。容器负责提供公用的服务, 例如目录服务、事务管理、安全性、资源缓冲池以及容错性。但这里值得注意的是, EJB 并不是实现 Java EE 的唯一途径。正是由于 Java EE 的开放性, 使得有的厂商能够以一种和 EJB 平行的方式来达到同样的目的。

4. RMI (Remote Method Invoke)

正如其名字所表示的那样, RMI 协议采用调用远程对象上的方法。它使用了序列化方式在客户端和服务端传递数据。RMI 是一种被 EJB 使用的更底层的协议。

5. Java IDL/CORBA

在 Java IDL 的支持下, 开发人员可以将 Java 和 CORBA 集成在一起。他们可以创建 Java 对象并使之可在 CORBA ORB 中展开, 或者他们还可以创建 Java 类并作为和其他 ORB 一起展开的 CORBA 对象的客户。后一种方法提供了另外一种途径, 通过它 Java 可以被用于将新的应用和旧的系统相集成。

6. JSP (Java Server Pages)

JSP 页面由 HTML 代码和嵌入其中的 Java 代码所组成。服务器在页面被客户端所请求以后对这些 Java 代码进行处理, 然后将生成的 HTML 页面返回给客户端的浏览器。

7. Java Servlet

Servlet 是一种小型的 Java 程序, 它扩展了 Web 服务器的功能。作为一种服务器端的应用, 当被请求时开始执行, 这和 CGI Perl 脚本很相似。Servlet 提供的功能大多与 JSP 类似, 不过实现的方式不同。JSP 通常是大多数 HTML 代码中嵌入少量的 Java 代码, 而 Servlets 全部由 Java 写成并且生成 HTML。

8. XML (Extensible Markup Language)

XML 是一种可以用来定义其他标记语言的语言。它被用来在不同的商务过程中共享数据。

XML 的发展和 Java 是相互独立的, 但是, 它和 Java 具有相同目标即平台独立性。通过将 Java 和 XML 的组合, 可以得到一个完美的具有平台独立性的解决方案。

9. JMS (Java Message Service)

JMS 是用于和面向消息的中间件相互通信的应用程序接口 (API)。它既支持点对点的域, 有支持发布/订阅 (Publish/Subscribe) 类型的域, 并且提供对下列类型的支持: 经认可的消息传递, 事务型消息的传递, 一致性消息和具有持久性的订阅者支持。JMS 还提供了另一种方式来对应用与旧的后台系统相集成。

10. JTA (Java Transaction Architecture)

JTA 定义了一种标准的 API, 应用系统由此可以访问各种事务监控。

11. JTS (Java Transaction Service)

JTS 是 CORBA OTS 事务监控的基本的实现。JTS 规定了事务管理器的实现方式。该事务管理器是在高层支持 Java Transaction API (JTA) 规范, 并且在较底层实现 OMG OTS Specification 的 Java 映像。JTS 事务管理器为应用服务器、资源管理器、独立的应用以及通信资源管理器提供了事务服务。

12. JavaMail

JavaMail 是用于存取邮件服务器的 API, 它提供了一套邮件服务器的抽象类。不仅支持 SMTP 服务器, 也支持 IMAP 服务器。

13. JAF (Java Beans Activation Framework)

JavaMail 利用 JAF 来处理 MIME 编码的邮件附件。MIME 的字节流可以被转换成 Java 对象, 或者转换自 Java 对象。大多数应用都可以不需要直接使用 JAF。

1.6 Java EE 相关名词

容器: 充当中间件的角色。

Web 容器: 给处于其中的应用程序组件 (JSP, Servlet) 提供一个环境, 使 JSP, Servlet 直接与容器中的环境变量接口交互, 不必关注其他系统问题。主要由 Web 服务器来实现。例如: Tomcat, Weblogic, Websphere 等。该容器提供的接口严格遵守 Java EE 规范中的 Web