中小型信息管理系统开发实例系列丛书

Java 数据库系统开发实例导航

求是科技 编著

人民邮电出版社

需要全本请访问 www.ertongbook.com

内 容 简 介

本书安排了利用 Java 技术开发的 6 个典型的小型信息管理系统，分别是“基于 Struts 的客户资料管理系统”、“基于 Struts 的人事管理系统”、“办公日志系统”、“电子政务档案管理系统”、“餐饮行业进销存系统”、“手机短信数据采集系统”，其中“餐饮行业进销存系统”系统采用了 C/S 结构，其余案例采用了 B/S 结构。

本书涉及到的信息系统的开发，采用了当前利用 Java 来设计数据库系统的主流、热门技术（JavaBean、Struts、手机短信技术等），并深入剖析了系统的开发思路、方法和技巧。帮助读者透彻掌握 Java 开发数据库系统的方法和步骤，能自行设计出具有实用价值的应用系统。

本书适合具备一定 Java 和数据库编程基础、正在从事或希望从事相关行业的应用开发的人员阅读。

中小型信息管理系统开发实例系列丛书

Java 数据库系统开发实例导航

- 编　　著　求是科技
  责任编辑　张立科
- 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
  读者热线：010-671940942
  北京汉魂图文设计有限公司制作
  北京顺义振华印刷厂印刷
  新华书店总店北京发行所经销
- 开本：787×1092　1/16
  印张：27
  字数：658 千字　　2004 年 5 月第 1 版
  印数：1 – 0 000 册　　2004 年 5 月北京第 1 次印刷

ISBN 7-115-12231-8/TP· 3945

定价：　.00 元（附光盘）

本书如有印装质量问题，请与本社联系　电话：（010）67129223

前　言

Java 是当前主流的开发工具之一，更是开发 Internet 应用系统的最佳利器之一。很多应用系统（尤其考虑平台无关性的时候）都要求用 Java 技术实现。

信息管理系统的应用范围较广，针对现有 Java 图书少有详细案例分析及设计的现状，本书安排了 6 个典型的利用 Java 开发的小型信息管理系统的案例，向读者剖析了利用 Java 进行系统设计和开发的特点、思路、方法和技巧，并详细讲述了各系统创建的过程。通过本书的学习，读者能方便地再现这些案例系统。

本书案例的介绍过程由教学目标与案例预览、系统分析与设计、分析与创建数据库和系统的具体创建和实现过程等几部分组成。读者只要顺序阅读，就能轻松再现各案例系统并理解相关技术，对相应的行业功能设计也有所掌握。更方便读者在本书案例基础上进一步加以完善和优化，应用于实际的应用中去。

为了便于读者学习，本书第 1 章主要介绍了 Struts 的相关知识，包括 Struts 的体系结构、配置文件、主要应用“类”、错误处理和标记库等，让读者了解和掌握 Struts 的工作原理、设计架构和开发手段，也为本书后面基于 Struts 而设计的案例做铺垫。

从第 2 章到第 7 章是本书的精华部分，即为 6 个典型案例程序的介绍，它们分别是“基于 Struts 的客户资料管理系统”、“基于 Struts 的人事管理系统”、“办公日志系统”、“电子政务档案管理系统”、“餐饮行业进销存系统”、“手机短信数据采集系统”。这些案例在功能和数据结构的设计上基本符合实际的需求，至少实现了最基本的功能，同时也很好地应用了相关技术，相信读者从技术和功能两个方面都能有所收获。

本书主要由求是科技策划，参与本书编写的人员有曹广鑫、彭涛、杨岚、刘静等。以上人员对本书的顺利完成付出了辛勤的汗水和心血，在此一并表示感谢。

由于时间仓促，加之水平有限，书中的缺点和不足之处在所难免，敬请读者批评指正。

欢迎读者访问求是科技网站http://www.cs-book.com，提出您宝贵的意见和建议。对于您遇到的问题，我们也将予以尽快解答。

编者

目　　录

第 1 章　Struts 基础和应用 ····· 1
1.1　为什么采用 Struts 构建 Web 应用程序 ····· 1
1.2　Struts 的简介 ····· 1
1.3　Struts 的体系结构 ····· 2
1.4　Struts 配置文件 ····· 3
1.4.1　配置全局转发 ····· 3
1.4.2　配置 ActionMapping ····· 4
1.4.3　配置 ActionForm Bean ····· 5
1.4.4　配置数据源 ····· 5
1.5　ActionServlet 类 ····· 6
1.5.1　配置 ActionServlet 类 ····· 7
1.5.2　ActionServlet 方法 ····· 8
1.6　ActionMapping 类 ····· 8
1.7　Action 类 ····· 8
1.8　ActionForm 类 ····· 9
1.9　ActionForward 类 ····· 10
1.10　错误处理 ····· 10
1.11　Struts 标记库 ····· 11
1.11.1　Bean 标记 ····· 12
1.11.2　逻辑标记 ····· 15
1.11.3　HTML 标记 ····· 18
1.11.4　模板标记 ····· 24

第 2 章　基于 Struts 的客户资料管理系统 ····· 26
2.1　教学目标与案例预览 ····· 26
2.1.1　教学目标 ····· 26
2.1.2　配置运行 ····· 26
2.1.3　系统预览 ····· 28
2.2　系统分析与设计 ····· 29
2.2.1　需求分析 ····· 29
2.2.2　模块设计 ····· 30
2.3　分析与创建数据库 ····· 31
2.3.1　创建 Oracle 数据库 ····· 31
2.3.2　创建 Struts 的 DataSource ····· 32
2.3.3　调用 DataSource 获取数据库连接 ····· 32
2.4　创建 JBuilder 工程（Project）和 Web 应用程序 ····· 32

2.5 设计与实现“国家名称管理”功能 ····· 33
2.5.1 设计与实现“国家”视图（View） ····· 33
2.5.2 设计与实现“国家”模型（Model） ····· 41
2.5.3 设计与实现“国家”控制层（Control） ····· 42
2.5.4 设计和实现“国家”业务逻辑（Data Access Object） ····· 51
2.6 设计与实现省份管理 ····· 55
2.6.1 设计与实现省份视图（View） ····· 55
2.6.2 设计和实现省份模型（Model） ····· 60
2.6.3 设计和实现省份控制层（Control） ····· 61
2.6.4 设计和实现省份封装业务逻辑 ····· 68
2.7 设计与实现城市管理 ····· 72
2.7.1 设计与实现城市视图（View） ····· 72
2.7.2 设计和实现城市模型（Model） ····· 78
2.7.3 设计和实现城市控制层（Control） ····· 79
2.7.4 设计和实现城市封装业务逻辑 ····· 85
2.8 设计与实现客户管理 ····· 90
2.8.1 设计与实现客户视图（View） ····· 90
2.8.2 设计和实现客户模型（Model） ····· 96
2.8.3 设计和实现客户控制层（Control） ····· 97
2.8.4 设计和实现客户封装业务逻辑 ····· 104
2.9 编译与部署系统 ····· 109
2.10 本章小结 ····· 109

第 3 章 基于 Struts 的人事管理系统 ····· 110
3.1 教学目标与案例预览 ····· 110
3.1.1 教学目标 ····· 110
3.1.2 配置运行 ····· 110
3.1.3 系统预览 ····· 112
3.2 系统分析与设计 ····· 113
3.2.1 需求分析 ····· 113
3.2.2 模块设计 ····· 114
3.3 分析与创建数据库 ····· 115
3.3.1 创建 Oracle 数据库 ····· 115
3.3.2 创建 Struts 的 DataSource ····· 116
3.3.3 调用 DataSource 获取数据库连接 ····· 116
3.4 创建 JBuilder 工程（Project）和 Web 应用程序 ····· 117
3.5 设计与实现公司管理 ····· 117
3.5.1 设计与实现公司视图（View） ····· 117
3.5.2 设计和实现公司模型（Model） ····· 120
3.5.3 设计和实现公司控制层（Control） ····· 121

需要全本请访问 www.ertongbook.com

3.5.4 设计和实现公司业务逻辑（Data Access Object） ····· 127
3.6 设计与实现职位管理 ····· 131
3.6.1 设计与实现职位视图（View） ····· 131
3.6.2 设计和实现职位模型（Model） ····· 134
3.6.3 设计和实现职位控制层（Control） ····· 135
3.6.4 设计和实现职位逻辑 ····· 141
3.7 设计与实现工作类型管理 ····· 145
3.7.1 设计与实现工作类型视图（View） ····· 145
3.7.2 设计和实现工作类型模型（Model） ····· 148
3.7.3 设计和实现工作类型控制层（Control） ····· 148
3.7.4 设计和实现工作类型业务逻辑 ····· 154
3.8 设计与实现部门管理 ····· 158
3.8.1 设计与实现部门视图（View） ····· 158
3.8.2 设计和实现部门模型（Model） ····· 162
3.8.3 设计和实现部门控制层（Control） ····· 162
3.8.4 设计和实现部门业务逻辑 ····· 168
3.9 设计与实现员工管理 ····· 172
3.9.1 设计与实现员工视图（View） ····· 172
3.9.2 设计和实现员工模型（Model） ····· 183
3.9.3 设计和实现员工控制层（Control） ····· 186
3.9.4 设计和实现员工业务逻辑 ····· 194
3.10 编译与部署系统 ····· 200
3.11 本章小结 ····· 200

第 4 章 办公日志系统 ····· 201
4.1 教学目标与案例预览 ····· 201
4.1.1 教学目标 ····· 201
4.1.2 配置运行 ····· 201
4.1.3 系统预览 ····· 205
4.2 系统分析与设计 ····· 208
4.2.1 需求分析 ····· 208
4.2.2 模块设计 ····· 208
4.3 分析与创建数据库 ····· 212
4.3.1 设计数据库结构 ····· 212
4.3.2 创建并配置数据库 ····· 213
4.4 创建办公日志管理系统 ····· 215
4.4.1 数据访问对象 DbADO.java ····· 216
4.4.2 业务对象模型 OffLogBean.java ····· 219
4.4.3 前台页面显示 ····· 226
4.4.4 web 配置文件 web.xml ····· 242

4.5 编译和部署系统 ····· 242
4.6 本章小结 ····· 243

第 5 章 电子政务档案管理系统 ····· 244
5.1 教学目标与案例预览 ····· 244
5.1.1 教学目标 ····· 244
5.1.2 配置运行 ····· 244
5.1.3 系统预览 ····· 247
5.2 系统分析与设计 ····· 250
5.2.1 需求分析 ····· 250
5.2.2 模块设计 ····· 251
5.3 分析与创建数据库 ····· 254
5.3.1 设计数据库结构 ····· 254
5.3.2 创建并配置数据库 ····· 256
5.4 编写管理系统的 JavaBean ····· 256
5.4.1 实现用户管理功能的 JavaBean——UserMan.java ····· 257
5.4.2 实现档案管理各功能的 JavaBean——DocMan.java ····· 262
5.4.3 实现档案库管理各功能的 JavaBean——DocBaseMan.java ····· 270
5.4.4 实现部门管理各功能的 JavaBean——DepartmentMan.java ····· 276
5.4.5 实现借阅管理各功能的 JavaBean——BorrowMan.java ····· 280
5.4.6 实现系统日志和消息管理功能的 JavaBean——SysLog.java ····· 287
5.4.7 实现内码转换功能的 JavaBean——ChangeEncoding.java ····· 289
5.5 设计与编写 JSP 页面 ····· 290
5.5.1 建立用户登录页面 ····· 290
5.5.2 建立主页面 ····· 291
5.5.3 建立用户管理页面 ····· 293
5.5.4 建立机构管理相关页面 ····· 294
5.5.5 建立档案管理相关页面 ····· 297
5.5.6 建立档案库管理相关页面 ····· 304
5.5.7 建立借阅管理相关页面 ····· 306
5.5.8 建立档案查询相关页面 ····· 312
5.5.9 建立“我的档案”功能相关页面 ····· 316
5.5.10 建立操作执行页面 action.jsp ····· 317
5.5.11 建立公共 Javascript 脚本文件 tools.js ····· 322
5.6 编译与部署系统 ····· 323
5.7 本章小结 ····· 323

第 6 章 餐饮行业进销存系统 ····· 325
6.1 教学目标与案例预览 ····· 325
6.1.1 教学目标 ····· 325

6.1.2 配置运行 ····· 325
6.1.3 系统预览 ····· 326
6.2 系统分析与设计 ····· 327
6.2.1 需求分析 ····· 327
6.2.2 模块设计 ····· 327
6.3 分析与创建数据库 ····· 328
6.3.1 设计数据库结构 ····· 329
6.3.2 创建并配置数据库 ····· 332
6.4 在 JBuilder 9 中建立开发环境 ····· 332
6.4.1 设置数据库驱动程序 ····· 333
6.4.2 设置数据库连接 ····· 334
6.5 系统模块创建 ····· 335
6.5.1 创建用户登录验证模块 ····· 335
6.5.2 创建系统主界面 ····· 341
6.5.3 创建系统管理功能模块 ····· 343
6.5.4 创建营业设置模块 ····· 349
6.5.5 创建前台营业模块 ····· 352
6.5.6 创建营业信息分析模块 ····· 363
6.5.7 创建库存管理界面 ····· 367
6.6 本章小结 ····· 369

第 7 章 手机短信数据采集系统 ····· 370
7.1 教学目标与案例预览 ····· 370
7.1.1 教学目标 ····· 370
7.1.2 配置运行 ····· 370
7.1.3 系统预览 ····· 370
7.2 系统分析与设计 ····· 374
7.2.1 需求分析 ····· 374
7.2.2 模块设计 ····· 375
7.3 分析与创建数据库 ····· 375
7.3.1 创建 Oracle 数据库 ····· 375
7.3.2 创建 Struts 的 DataSource ····· 376
7.3.3 GSM 简介 ····· 377
7.4 创建 JBuilder 工程（Project）和 Web 应用程序 ····· 377
7.5 设计与实现手机用户管理 ····· 377
7.5.1 设计与实现用户视图（View） ····· 377
7.5.2 设计和实现用户模型（Model） ····· 381
7.5.3 设计和实现手机用户控制层（Control） ····· 381
7.5.4 设计和实现用户业务逻辑（Data Access Object） ····· 387
7.6 设计与实现收发短信管理 ····· 391

7.6.1 设计与实现收发短信视图（View） ····· 391
7.6.2 设计和实现收发短信模型（Model） ····· 396
7.6.3 设计和实现收发短信控制层（Control） ····· 397
7.6.4 设计和实现收发短信逻辑 ····· 403
7.7 设计与实现促销管理 ····· 407
7.7.1 设计与实现促销视图（View） ····· 407
7.7.2 设计和实现促销模型（Model） ····· 410
7.7.3 设计和实现促销控制层（Control） ····· 411
7.7.4 设计和实现促销业务逻辑 ····· 417
7.8 编译与部署系统 ····· 421
7.9 本章小结 ····· 421

第 1 章　Struts 基础和应用

本章目的是向读者介绍 Struts 的优点、Struts 的配置和使用方法、Struts 在软件开发过程中的设计和实现以及 Tiles 组件的使用等知识点，其他信息请读者参考http://jakarta.apache.org/Struts。

1.1　为什么采用 Struts 构建 Web 应用程序

Struts 属于 Jakarta 的一个项目，是目前流行最广的 MVC 体系架构，它可以在 Web 应用程序中，使用 JavaServer Pages（JSP）和 Servlets。Struts 的出现解决了由 JSP 或由 Servlets 开发 Web 应用程序的所遇到的问题。大多数情况下，可以使用 Servlets 生成 HTML 页面，传送给客户端，但这样所有的代码都由 Servlets 来写，其代码比较麻烦，而且维护起来很困难。另外，使用传统的 JSP 来生成 HTML 页面，业务逻辑和显示逻辑混杂在一起，很多 Java 代码和 HTML 代码混在一起，代码维护量大，开发效率低下。

目前，很多公司采用 Model1 架构，就是用 JSP 负责显示逻辑和业务逻辑，将用户请求的页面返回到客户端。鉴于 JSP 开发技术上手比较容易，所以很多公司采用 Model1 架构来开发 Web 应用程序。

JSP 和 Servlets 属于目前 Java 两个重要的技术，在开发 Web 应用程序中，有很广泛的应用。JSP 负责针对用户的显示逻辑，Servlets 负责针对用户的业务逻辑，这两者的结合，被称为 Model2 架构或者 Model-View-Control（MVC）架构。

在开发 Web 应用程序的时候，使用 Model1 架构和 Struts 的不同之处在于针对用户的请求程序是如何处理的。基于 Model1 架构，所有的请求和处理都是通过 JSP 页面，显示逻辑和业务逻辑没有分开。而 Struts 采用了 JSP 和 Servlets 的长处，JSP 属于 Struts 的 View（视图），它负责显示逻辑，将客户所请求的结果返回到客户端。Servlets 属于 Struts 的 Control（控制器），它负责处理客户的请求，并将请求的结果返回视图，显示给读者。

下面，来讲述 Struts 的体系架构和相关技术。

1.2　Struts 的简介

读者可以从 Struts 的官方网站 http://jakarta.apche.org 下载到最新的压缩文件-jakarta-Struts-1.1.zip，该文件包含 3 个目录：contrib、lib 和 webapps。在 contrib 目录中所包含的文件在 lib 和 webapps 目录中都存在，这里只介绍 lib 和 webapps 目录。在 lib 目录中有使用 Struts 创建应用程序时所需的文件，如表 1-1 所示。

表 1-1　　Struts 压缩包下 lib 目录所包含的文件

文件	描述
*.jar	包含 Struts 所有类的 Java 类包，以及支持 Struts 运行的 Java 类包
*.tld	标记库描述器文件，描述了多个标记库的自定义标记。在 Web 应用程序开发过程中，要拷贝到 WEB-INF\LIB 目录下

webapps 目录下的文件如表 1-2 所示。

表 1-2　　Webapps 目录下的文件

Web 应用程序	描述
Struts-blank.war	一个简单的 web 应用程序
Struts-documentation.war	包含 Struts 站点上所有 Struts 文档
Struts-example.war	Struts 很多特性的示范
Struts-exercisetaglib.war	介绍主要用于对自定义标签库进行增加而使用的测试页，但也可以示范如何使用 Struts 标记
Struts-template.war	包含 Struts 模板标记的介绍和范例
Struts-upload.war	一个简单的例子，示范如何使用 Struts 框架上传文件
Struts-validator.war	关于 JavaScript 验证的例子
tiles-documentation.war	关于 Web 应用程序使用 Tiles 组件布局的例子

Struts 1.1 版本比以前增加很多内容，系统架构和性能比以前更加完善，开发过程中会更加方便。

1.3　Struts 的体系结构

Struts 通过 ActionServlet 实现了 Model2 架构，即是 MVC 架构。Model 表示模型，表示一个应用系统的数据并且包含访问和管理该数据的逻辑。所有属于应用系统持久状态的数据都应该保存于模型的对象里。一个模型的接口提供了访问和更新模型状态的、执行封装在模型中的复杂进程的方法。模型服务被控制器访问，用于查询或使模型的状态发生变化。当状态发生变化时，模型会通报给视图。视图的作用是表现模型的状态。表述语句封装于视图中，因此模型数据可以适合多种不同的终端。当模型中的变化传达到视图的时候，视图会修改自己。视图将用户输入传递到控制器。控制器的任务是获取并翻译用户输入到动作并由模型执行。控制器的任务还有根据用户输入和执行结果选择下一个视图。在基于 J2EE 的应用系统中，MVC 结构用于分离由 Java Beans 或 EJB 表示的事物层及 JSP 表示的表现层，中间经由 Servlet 的控制器。然而，控制器设计必须适合各种不同类型终端的输入，包括来自 Web 终端的 HTTP 请求、无线终端的 WML、供应商与业务伙伴的 XML 的文档等。对于 HTTP 请求和任务范例，提交的 HTTP 请求被发送到一个控制中心，依次解释并委派该请求到适当的请求处理器。这也被作为第二类 MVC 结构。请求处理器与开发者提供的架构结合起来，以实现对与模型有关联的特定逻辑的请求。依靠这个交互的结果，控制器可以决定下一个视图以产生恰当的反应。这个控制器可以用来管理 JSP 页面和其他表现层之间的流程。Struts 通过使用 ActionForward 和 ActionMapping 来实现 MVC 的视图（View）。JSP 可以引用一个逻辑目标。控制器组件在运行时提供准确的 URL。

表1-3列出了Struts的核心类，及对应的MVC组件功能的描述。

需要全本请访问 www.ertongbook.com

表 1-3　　Struts 的核心类

类名称	描述
ActionServlet	控制器，接受用户请求和状态改变，以及发出视图选择
ActionClass	包含事务逻辑
ActionForm	显示模块数据
ActionMapping	帮助控制器将请求映射到操作
ActionForward	用来指示操作转移的对象
ActionError	用来存储和回收错误
Struts 标记库	可以减轻开发显示层次的工作

除了这些核心类，Struts 使用一些配置文件和视图助手（View Helpers）来沟通控制器和模型。表 1-4 列出了 Struts 配置文件和描述了它们在架构中的角色。

表 1-4　　Strtuts 配置文件

文件	目的
ApplicationResources.properties	存储本地化信息和标签，以使应用可以国际化，支持不同的语言
Struts-config.xml	存储控制器对象的默认配置，包括模型支持的用户指向，状态改变，状态查询

为将 Struts 配置数据提供给视图，框架以 JSP 标签的形式提供了大量的助手类，如表 1-5 所示。

表 1-5　　Strtus 标记库

标记库描述符	目的
Struts-html.tld	扩展 HTML Form 的 JSP 标记
Struts-bean.tld	扩展处理 JavaBean 的 JSP 标记
Struts-logic.tld	扩展测试属性值的 JSP 标记

实践中，控制器与视图的交互通过请求，会话以及Servlet平台提供的应用上下文进行。

1.4　Struts 配置文件

Struts-config.xml 配置文件包含了 ActionServlet 需要用来处理对应用请求的详细信息。文件存储在<BaseDirectory>/webapps/crm/WEB-INF/目录下。配置文件可以定义为：

- 全局转发。
- ActionMapping 类。
- ActionForm bean。
- 数据源。

1.4.1　配置全局转发

全局转发用来在 JSP 页之间创建逻辑名称映射。转发都可以通过对调用操作映射的实例来获得，例如：

```
actionMapping.findForward(″failure″);
```

全局转发的实现代码如下，属性如表 1-6 所示。

```
<!--配置全局转发，名称为 Failure，路径为/login.jsp-->
<global-forwards>
<forward     name="failure"    path="/login.jsp"/>
```

```
</global-forwards>
```

表 1-6　　全局转发

属性	描述
Name	全局转发的名字
Path	与目标 URL 的相对路径

1.4.2　配置 ActionMapping

ActionMapping 对象帮助进行框架内部的流程控制，它们可将请求 URL 映射到 Action 类，并且将 Action 类与 ActionForm bean 相关联。ActionServlet 在内部使用这些映射，并将控制转移到特定 Action 类的实例。所有 Action 类使用 perform()方法实现特定应用程序代码，返回一个 ActionForward 对象，其中包括响应转发的目标资源名称，属性如表 1-7 所示，具体代码如下：

```
<!--
配置 ActionMapping，路径为/Nation/add ，参数为 ADD，类型为 com.landsoft.personal.controller.
NationAction，当 ADD 配置成功的时候，Forward 名称为 Success，路径为 manager.Nation.entry。
-->
<action-mappings>
<action path="/Nation/add" parameter="ADD" type="com.landsoft.personal.controller.NationAction"
        scope="request" name="nationForm" validate="false">
        <forward name="success" path="manager.Nation.entry"/>
    </action>
</action-mappings>
```

表 1-7　　配置 ActionMapping

属性	描述
Path	Action 类的相对路径
Name	与本操作关联的 Action bean 的名称
Type	连接到本映射的 Action 类的全称（可有包名）
Scope	ActionForm bean 的作用域（请求或会话）
Prefix	用来匹配请求参数与 bean 属性的前缀
Suffix	用来匹配请求参数与 bean 属性的后缀
attribute	作用域名称
className	ActionMapping 对象的类的完全限定名默认的类是 org.apache.Struts.action.ActionMapping
input	输入表单的路径，指向 bean 发生输入错误必须返回的控制
unknown	设为 true，操作将被作为所有没有定义的 ActionMapping 的 URL 的默认操作
Validate	设置为 true，则在调用 Action 对象上的 perform()方法前，ActionServlet 将调用 ActionForm bean 的 validate()方法来进行输入检查

通过<forward>元素，可以定义资源的逻辑名称，该资源是Action类的响应要转发的目标，如表1-8所示。

表 1-8　　配置 F

属性	描述
Id	ID
ClassName	ActionForward 类的完全限定名，默认是 org.apache.Struts.action.ActionForward
Name	操作类访问 ActionForward 时所用的逻辑名

续表

属性	描述
Path	响应转发的目标资源的路径
redirect	若设置为 true，则 ActionServlet 使用 sendRedirect()方法来转发资源

1.4.3　配置 ActionForm Bean

ActionServlet 使用 ActionForm 来保存请求的参数，这些 bean 的属性名称与 HTTP 请求参数中的名称相对应，控制器将请求参数传递到 ActionForm bean 的实例，然后将这个实例传送到 Action 类，属性如表 1-9 所示，具体代码如下：

```
<!---配置 Web 应用程序所需要的 ActionForm Bean->
<form-beans>
    <form-bean type="com.landsoft.personal.controller.NationForm" name="nationForm"/>
    <form-bean type="com.landsoft.personal.controller.BusinessForm" name="businessForm"/>
    <form-bean type="com.landsoft.personal.controller.TypeofworkForm" name="typeofworkForm"/>
    <form-bean type="com.landsoft.personal.controller.DeptForm" name="deptForm"/>
    <form-bean type="com.landsoft.personal.controller.PersonalForm" name="personalForm"/>
  </form-beans>
```

表 1-9　　配置 ActionForm Bean

属性	描述
Id	ID
className	ActionForm bean 的完全限定名，默认值是 org.apache.Struts.action.ActionFormBean
Name	表单 bean 在相关作用域的名称，这个属性用来将 bean 与 ActionMapping 进行关联
Type	类的完全限定名

1.4.4　配置数据源

用<data-sources>元素可以定义多个数据源，如表1-10所示。

表 1-10　　配置数据源

属性	描述
Id	ID
Key	Action 类使用这个名称来寻找连接
Type	实现 JDBC 接口的类的名称

下面属性需要<set-property>元素定义，<data-source>元素属性如表 1-11 所示。代码如下：

```
<!--配置 Struts 的数据源-->
<data-sources>
        <data-source>
<!--配置是否自动提交-->
          <set-property property="autoCommit" value="false"/>
<!--配置数据源的描述-->
           <set-property property="description" value="Database for mobile of System"/>
<!--配置数据源的 DriverClass-->
           <set-property property="driverClass" value="sun.jdbc.odbc.JdbcOdbcDriver"/>
<!--配置数据源的最大连接数-->
           <set-property property="maxCount" value="5"/>
<!--配置数据源的最小连接数-->
          <set-property property="minCount" value="1"/>
<!--配置数据源的 URL-->
```

```
            <set-property property="url" value="jdbc:odbc:erp"/>
    <!--配置数据源的用户-->
            <set-property property="user" value="sa"/>
    <!--配置数据源用户的密码-->
            <set-property property="password" value="password"/>
        </data-source>
        </data-sources>
```

表 1-11　　配置数据源属性值

属性	描述
desciption	数据源的描述
autoCommit	数据源创建的连接所使用的默认自动更新数据库模式
driverClass	数据源所使用的类，用来显示 JDBC 驱动程序接口
loginTimeout	数据库登录时间的限制，以秒为单位
maxCount	最多能建立的连接数目
minCount	要创建的最少连接数目
password	数据库访问的密码
readOnly	创建只读的连接
User	访问数据库的用户名
url	JDBC 的 URL

通过指定关键字名称，Action 类可以访问数据源，实例代码如下：

```
//从数据源取得数据库连接
DataSource ds = (DataSource) servlet.getServletContext().getAttribute(Action.DATA_SOURCE_KEY);
//调用数据库连接
DeptDAO deptDAO = new DeptDAO(ds);
```

1.5　ActionServlet 类

Struts 框架的控制器（Control）是通过 Struts 的 org.apache.Struts.action.ActionServlet 类实现的，这个类是 javax.servlet.http.HttpServlet 类的扩展。

Struts 的控制器的基本功能是：

- 获取用户的 Http 请求。
- 将用户的请求映射到相应的 Action 类，初始化实例并缓存。
- 创建 ActionForm bean 实例（从 Struts-config.xml 文件调用 ActionForm 配置文件），然后将请求过程移植到 bean。
- 调用 Action 实例的 perform()方法并将 ActioForm bean、Action Mapping 对象、request 和 response 对象传给它。

例如：

```
public ActionForward execute(ActionMapping mapping, ActionForm form,
        HttpServletRequest request, HttpServletResponse response) {
//声明 ActionForward
    ActionForward myforward = null;
//取得当前的 Action
    String myaction = mapping.getParameter();
//判断用户是否单击“取消”按钮
    if (isCancelled(request)) {
      if (_log.isInfoEnabled()) {
```

```
                _log.info(" [Dept] " + mapping.getAttribute() + " - action was cancelled");
            }
            return mapping.findForward("cancel");
        }
        if (_log.isInfoEnabled()) {
            _log.info(" [Dept] action: "+myaction);
        }
    //根据 myAction 的值，来执行不同的方法
        if ("".equalsIgnoreCase(myaction)) {
            myforward = mapping.findForward("failure");
        } else if ("VIEW".equalsIgnoreCase(myaction)) {
          myforward = performView(mapping, form, request, response);
        } else if ("EDIT".equalsIgnoreCase(myaction)) {
            myforward = performEdit(mapping, form, request, response);
        } else if ("ADD".equalsIgnoreCase(myaction)) {
            myforward = performAdd(mapping, form, request, response);
        } else if ("SAVE".equalsIgnoreCase(myaction)) {
            myforward = performSave(mapping, form, request, response);
        } else if ("REMOVE".equalsIgnoreCase(myaction)) {
            myforward = performRemove(mapping, form, request, response);
        } else if ("TRASH".equalsIgnoreCase(myaction)) {
            myforward = performTrash(mapping, form, request, response);
        } else if ("LIST".equalsIgnoreCase(myaction)) {
            myforward = performList(mapping, form, request, response);
        } else {
            myforward = mapping.findForward("failure");
        }
        return myforward;
    }
```

- perform 返回一个 ActionForword 对象，可以调用配置文件中的 ActionMapping 类调用配置文件并显示 JSP 页面。

1.5.1　配置 ActionServlet 类

需要在 web.xml 中声明 ActionServlet，并且将它配置成启动时加载，表 1-12 为可以配置的初始化参数。

表 1-12　　配置 ActionServlet

参数	默认值	描述
application	null	应用程序的资源集合的类
bufferSize	4096	文件上传的缓冲区大小
config	/WEB-INF/Struts-config.xml	配置文件的位置和名称
content	Text/html	默认的内容类型
debug	0	程序调试的级别
detail	0	程序调试细节的级别
factory	null	消息资源工厂，用于国际化中解释消息资源
formBean	org.apache.Struts.action.ActionFormBean	封装 ActionForm bean 信息的类的名称
forward	org.apache.Struts.action.ActionForward	封装 ActionForward 对象信息的类的名称
locale	true	为 true，将在用户会话中存储一个本地对象
mapping	org.apache.Struts.action.ActionForward	封装 ActionMapping 信息的类的名称

续表

参数	默认值	描述
maxFileSize	250M	上传文件的最大尺寸
multipartClass	org.apache.Struts.action.ActionForward	处理多部分请求的类的名称
noCache	False	HTTP 标头是否要设置为禁止缓存
Null	True	设置为 true，对于无效的信息关键字将返回 null
tempDir	作为一个 servlet 参数提供给程序的工作目录	处理下载文件是使用的临时工作目录
validate	True	是否使用新格式的配置文件
vallidating	True	是否对配置文件进行有效性分析

当系统首次处理用户请求的 URL 时，ActionServlet 将适当的 Action 类进行实例化，然后 ActionServlet 在 Action 类实例中以 servlet 为变量名存储一个引用。当被实例化后，Action 类会被缓存以供程序以后调用。

1.5.2 ActionServlet 方法

ActinServlet 提供了能够被 Action 对象使用的方法。部分方法如下。

```
public void addFormBean(ActionFormBean formBean)
public void removeFormBean(ActionFormBean formBean)
public void addForward(ActionForward actionForward)
public void removeForward(ActionForward actionForward)
public void addMapping(ActionMapping actionMapping)
public void removeMapping(ActionMapping actionMapping)
```

1.6 ActionMapping 类

Struts 的控制器将用户请求映射到特定 Action 的相关信息存储在 ActionMapping 中，ActionServelt 将 ActionMapping 传送到 Action 类的 perform()方法，Action 将使用 ActionMapping 的 findForward()方法，返回一个指定名称的 ActionForward。若没有找到具体的 ActionForward，就返回一个 null。

```
public ActionForward findForward(String name)
```

可在映射中动态添加 ActionForward。

```
public void addForward(ActionForward forward)
```

可返回与映射关联的表单 bean。

```
public String getName()
```

可返回映射的属性域（会话或请求）。

```
public String getScope()
```

1.7 Action 类

Action 类真正实现应用程序的事务逻辑，它们负责处理请求。在收到请求后，ActionServlet 会进行如下操作：

- 选择适当的 Action。

需要全本请访问 www.ertongbook.com