

基于 SSH 框架与泛型的通用分页方法 设计与实现

黄美林, 马建华, 李 东

(总后勤部 后勤科学研究所, 北京 100071)

摘 要:为解决基于 J2EE 的 Web 应用中,信息的分页显示存在大量代码重复和实现过于复杂等问题,文中剖析分页处理过程存在的共性,同时分析目前广泛用于 Web 应用开发的开源框架 struts、spring、hibernate(简称 SSH)进行分析。通过 SSH 框架的整合以及 Java 泛型技术的应用,文章设计了一种通用的分页方法,并对该方法的实现原型进行设计,同时通过具体实例对实现过程涉及的通用类、核心属性和方法以及参数配置进行详细描述。通过在工程项目中的应用,验证了该方法在降低信息分页处理过程复杂度的同时,提高了分页代码的复用性,同时该方法具有通用高效、简单易用等特点。

关键词:SSH;泛型;通用分页

中图分类号:TP31

文献标识码:A

文章编号:1673-629X(2012)01-0067-05

Design and Implementation of Generic Pagination Means Based on SSH Architecture and Generics

HUANG Mei-lin, MA Jian-hua, LI Dong

(Institute of Logistics Sciences, General Logistics Department of PLA, Beijing 100071, China)

Abstract:To solve the problem that exists in information pagination of J2EE-based Web applications; large code duplication and too complicated to achieve, analyze the commonness pagination processing and the current widely used open source framework in Web application development struts, spring, hibernate(SSH for short). Through the integration of SSH and the Java generic technology, designed a universal paging method and its implementation prototype, while described the common class, core attributes and methods parameter configuration by means of concrete examples in detail. By the project application, this means reduces the pagination complexity of operating information and improves the repetition of pagination code greatly, while it has versatile, efficient and simple features.

Key words:SSH; generics; generic pagination

0 引 言

SSH 是基于 J2EE 架构的 Web 应用中广泛采用的开源框架,同时业务信息的分页显示是各类应用中都需要实现的一个基本功能。以往的分页处理,通常在各个业务逻辑单元中定义分页信息的存储变量,并根据传入参数实现分页处理,将处理结果置于待返回的分页结果存储变量中。由于分页显示过程大体相似,在所有的业务逻辑单元都进行具体实现的方式造成很多代码重复,同时以往的分页处理过程在层次划分上不够清晰,增加了分页显示实现的复杂性。为解决上述的问题,文中在整合 SSH 框架的同时,应用 JDK1.5 中新增的泛型技术,设计出一种通用的分页方法,并通

过标准信息分页处理实例对该方法的实现步骤和内容进行详细描述。

1 SSH 框架

1.1 概 述

SSH 框架将基于 J2EE 的 Web 应用划分为 Web 表现层、业务逻辑层、数据持久层^[1],通过三层框架的整合,有效地改善系统的伸缩性、可维护性、可扩展性及可重用性。其基本架构图如图 1 所示^[2]。

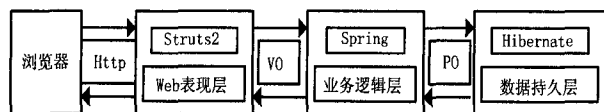


图 1 SSH 框架基本架构图

数据持久层负责业务数据的持久化存储,由 hibernate 的 O/R Mapping 技术实现。业务逻辑层负责业务对象转换传递、业务逻辑组织及事务控制等,由 spring Framework 实现业务 Beans 的依赖关系管理、运

收稿日期:2011-05-26;修回日期:2011-09-05

作者简介:黄美林(1981-),男,安徽怀宁人,硕士,工程师,研究方向为计算机软件与应用;马建华,工程师,研究方向为计算机软件与应用;李 东,硕士,高级工程师,研究方向为计算机软件与应用。

行时注入和事务控制等功能。Web 表现层由 struts 技术来实现,是所有业务逻辑的最终展现层,充当事务逻辑层的客户端,实现人机交互^[3]。

1.2 struts2

struts2 是一个基于 Sun J2EE 平台,为开发基于模型-视图-控制器(MVC)模式应用的开源框架,主要采用 Servlet 和 JSP 技术实现。体系概图如图 2 所示^[4]。

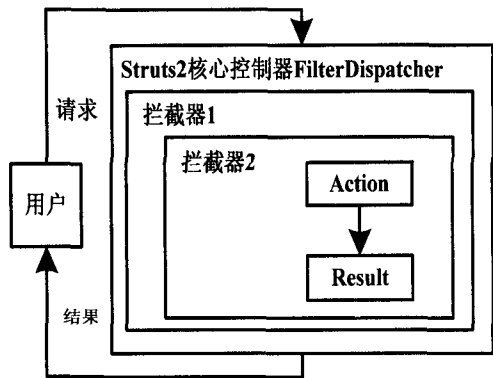


图 2 struts2 体系概图

struts2 的核心由控制器组成,包括核心控制器 FilterDispatcher 和拦截器^[5],其处理的过程分为五步:

- (1) 用户通过浏览器发出请求;
- (2) 核心控制器通过 FilterDispatcher 根据用户请求调用合适的 Action;
- (3) 拦截器自动对请求应用通用的处理功能;
- (4) 回调 Action 的 execute 方法,该方法获取用户请求参数,并调用业务逻辑;
- (5) Action 的 execute 方法根据执行结果返回 result 值,在 struts.xml 配置文件中,指定 result 值和视图资源之间的映射关系^[6]。

1.3 spring

spring 框架是由 Open Source 开发的一个优秀的多层 J2EE 系统框架,它为企业级应用提供了一个非常轻量级的解决方案,极大地降低了应用开发的难度与复杂度,提高系统开发速度和维护的灵活性。其体系结构如图 3 所示^[7]。

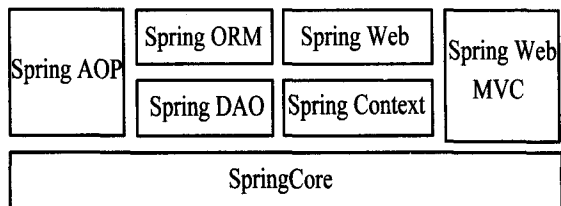


图 3 spring 体系架构图

spring 由七个模块组成^[8],其中 spring Core 提供 spring 框架的基本功能,主要组件是 BeanFactory,是工厂模式的实现,使用控制反转 (IOC) 模式将应用程序的配置和依赖性规范与实际的应用程序代码分开;spring Context 是一个配置文件,向 spring 框架提供上

下文信息;spring AOP 模块直接将面向方面的编程功能集成到 spring 框架中。使 spring 框架管理的任何对象支持 AOP;spring DAO 用于管理异常处理和不同数据库抛出的错误消息;spring ORM 用于提供对 JDO、hibernate 等框架集成的支持;spring Web 建立于 spring Context 之上,为基于 Web 的应用程序提供上下文信息;spring Web MVC 用于为开发基于 MVC 模式的 Web 应用提供支撑^[9]。

1.4 hibernate

hibernate 是一种 ORM 映射工具,它不仅提供了从 Java 类到数据表之间的映射,也提供了数据查询和恢复机制。相对于使用 JDBC 和 SQL 来手工操作数据库,使用 hibernate,可以大大减少操作数据库的工作量。hibernate 体系结构如图 4 所示^[10]。

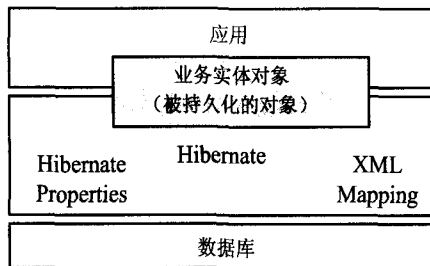


图 4 hibernate 体系结构图

应用程序为对象模型编写业务实体 Bean,通过 xml 映射文件或编写 Annotation 对其映射数据库的方式进行描述,最后使用 hibernate 提供的 API 操作业务实体 Bean 实现对其持久化操作^[11]。

2 通用分页设计与实现

2.1 Java 泛型

泛型是 JDK5.0 后引入的新特性;它的本质是参数化类型,也就是说所操作的数据类型被指定为一个参数。这种参数类型可以用在类、接口和方法的创建中,分别称为泛型类、泛型接口、泛型方法。泛型是在编译时检查类型安全,而且所有的强制转换都是自动和隐式的,从而提高代码的重用率。

2.2 通用页面存储类

通用分页信息存储类基于泛型进行设计,定义为 public class PageInfo<T>,类的实现主要包括四部分:

- (1) 页面处理私有成员,包括待分页显示记录总数 totalSize、总页数 totalPage,默认每页显示的记录数 pageSize,当前页页码 currentPage、第一条记录在总记录中的偏移量 offSet,实际显示的记录数 length;
- (2) 页面内容存储私有成员,用于存储当前页面显示具体内容,定义为 private List<T> pageList;
- (3) 私有成员的 getter 和 setter 方法,针对(1)和(2)中的私有成员,形成其 getter 和 setter 方法,用于辅

助实现 spring 的依赖注入;

(4)辅助完成分页功能的静态方法,包括计算总页数、偏移量、实际显示记录数以及页码值校正。

按照上述设计,实现该类的主要代码如下:

```
public class PageInfo<T> {
    private int pageSize;//默认分页显示的记录数
    private int currentPage;//当前页号
    private int totalSize;//分页实现总记录数
    private int totalPage;//分页显示总页数
    private int offset;//当前分页第一条记录在总记录
    中的偏移量
    private int length;//当前分页的记录数
    private List<T> pageList;//存储分页记录信息的
    List
    .....//成员变量的 getter 和 setter 方法
    //辅助函数,计算要显示记录的总页数
    public static int countTotalPage( int pageSize, int all-
    Row) {
        int totalPage = ( allRow % pageSize == 0 ) ? ( all-
    Row / pageSize ) : ( allRow / pageSize + 1 );
        return totalPage;
    }
    .....//其他静态辅助函数
}
```

2.3 通用分页过程设计

由于各类业务信息的分页显示功能相似,为达到各类业务 Bean 分页需求的通用处理,为避免重复编写代码,从以下两个方面进行设计。

(1)将 struts2、spring、hibernate 三个开源框架进行

整合,采用 struts2 实现前台页面展示,spring 实现业务逻辑转换处理,hibernate 实现与底层数据库的操作^[8];

(2)应用泛型技术,结合相关配置文件的参数化配置,将用户请求的业务 Bean 类型,注入分页处理过程,同时返回所注入业务 Bean 的分页处理结果。

基于上述两个方面,设计通用分页方法基本过程模型如图 5 所示:

基于该模型,具体业务 Bean 的分页处理过程分为以下五个步骤:

- (1)用户通过页面向 Web 服务器发出所请求的业务对象类型、查询条件和页码;
- (2)struts2 接收相关参数、条件信息后,转交业务信息处理 Action,其中业务信息处理 Action 类继承通用 Action 类 GenericAction;
- (3)spring 根据 Action 请求,调用 Action 中对应方法进行业务逻辑处理,处理过程中,如果涉及到数据库访问操作,则交由 hibernate 完成;
- (4)Action 将业务处理完毕后,将当前分页需要显示的业务对象信息存储到所继承的通用页面存储类成员变量 pageInfo 中;
- (5)struts2 将通用页面存储类中的 pageList 内容按需求,写入 jsp 页面后由 Web 服务器发送给用户端进行展现。

2.4 通用分页过程实现

要实现具体业务 Bean 的分页显示功能,需要完成以下三个方面的工作。

(1)通用类的实现。包括通用的分页信息存储类 PageInfo,通用 Action 类 GenericAction,这两个类完成对所有业务 Bean 分页处理过程的支持。其中 PageInfo

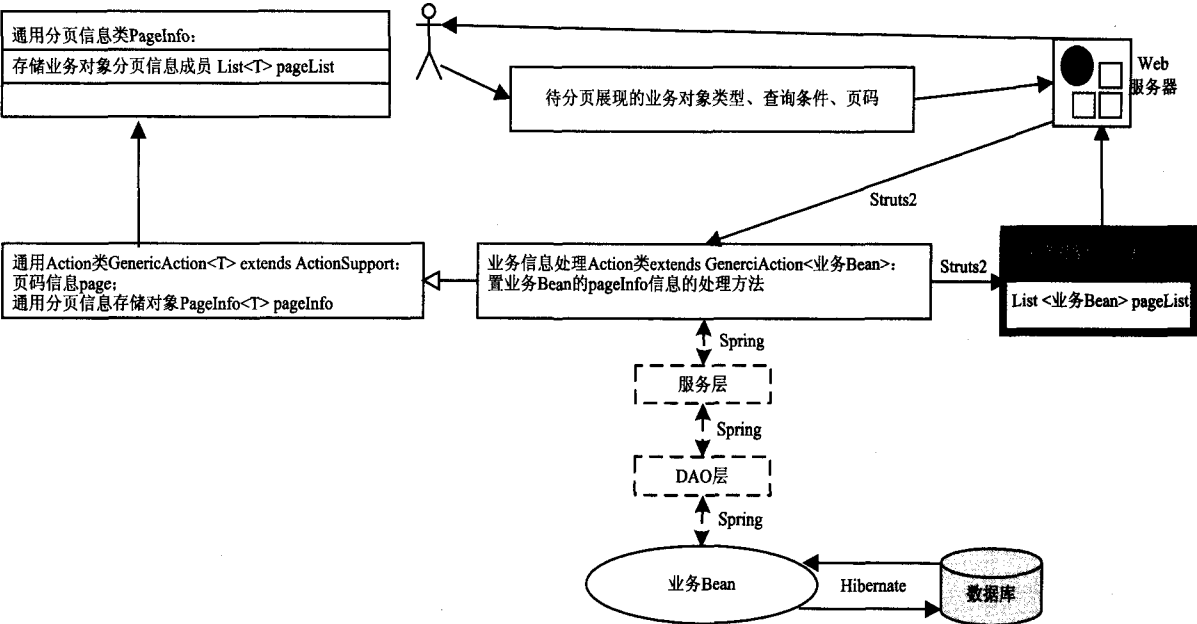


图 5 通用分页方法基本过程模型

类的实现见 2.2 节, GenericAction 用于定义所有业务处理 Action 需继承的变量和处理方法, 如通用处理服务、分页信息存储对象、页码等。

主要实现代码如下:

```
public class GenericAction<T> extends ActionSupport {
    protected GenericService<T> genericService;//根据 T 注入具体的 Service
    protected int page;//页码
    protected PageInfo<T> pageInfo;//通用的分页信息存储对象
    .....//成员变量的 get 和 set 方法
}
```

(2) 具体业务处理相关类的实现。包括具体业务 Bean 类和业务信息处理 Action 类, 这两个类完成对具体业务 Bean 的分页处理过程的支持。其中业务 Bean 类主要定义业务实体具体内容项, 并运用 Annotation 实现 Java 对象与数据库表、字段、关系等的映射。

以“标准信息”业务 Bean 为例, 实现代码如下:

```
@ Entity
@ Table( name = " BIAOZHUNXX ")//注解方式实现与数据库表的映射
public class BiaoZhunXXBean {
    private String ID;//唯一标识符
    private String zhongWenMC; //中文名称
    private String yingWenMC;//英文名称
    private String biaoZhunBH;//标准编号
    @ Id
    @ GeneratedValue( generator = " system-uuid " )
    @ GenericGenerator( name = " system-uuid ", strategy = " uuid " )//注解方式实现表字段及主键关系映射
    public String getID() {
        return ID;
    }
    .....// 成员变量的 get 和 set 方法, 并在 get 方法上运用 Annotation 实现与数据库的映射
}
```

业务信息处理 Action 主要实现业务逻辑的具体处理过程, 以标准信息查询业务为例, 通过传入 hsql 语句、默认分页记录数以及页码参数, 交由 hibernate 查询接口, 从数据库获取符合条件的记录, 实现代码如下:

```
public class BiaoZhunXXAction extends GenericAction<BiaoZhunXXBean> {
    public String list() {
        //将符合条件业务 Bean 的分页信息存储入 pageInfo 对象中
```

Info 对象中

```
pageInfo = this. genericService. queryForPage(
    searchSql, SystemInit. getPageSize(), page);
return SUCCESS;
}
.....//其他业务逻辑处理函数
}
```

(3) 对基于 SSH 框架 Web 应用的四个核心配置文件内容进行配置。其中 web. xml 中的配置实现 struts2、spring 框架功能的整合, 通过配置过滤器, 将所有页面的访问交由 struts2 框架处理, 通过配置监听器, 将业务逻辑的转发交由 spring 框架完成, 主要配置参数如下^[12]:

```
<filter>
<filter-name>struts2</filter-name>
<filter-class>org. apache. struts2. dispatcher. FilterDispatcher</filter-class>
</filter>
<filter-mapping>
<filter-name>struts2</filter-name>
<url-pattern>/ * </url-pattern>
</filter-mapping>
<listener>
<listener-class>org. springframework. web. context. ContextLoaderListener</listener-class>
</listener>
```

hibernate. cfg. xml 通过对数据库连接串、用户名、密码、驱动等参数的配置, 实现 web 应用与数据库连接, 通过类映射参数配置, 实现 java 类与数据库表之间的对应, 主要配置参数如下^[10]。

```
<hibernate-configuration>
<session-factory>
<property name = " connection. url " >jdbc: oracle: thin:@ localhost: 1521 : gpa</property>
<property name = " connection. username " >hqbz</property>
<property name = " connection. password " >hqbz</property>
<property name = " connection. driver_class " >oracle. jdbc. driver. OracleDriver</property>
<property name = " dialect " >org. hibernate. dialect. Oracle10gDialect</property>
<mapping class = " kys. zts. hqbz. bean. BiaoZhunXXBean " /> <!-- 标准信息类与数据库表映射声明 -->
</session-factory>
</hibernate-configuration>
```

struts.xml 中的配置实现页面跳转指向,下面以标准信息查询为例,BiaoZhunXXAction 中 list 方法处理完成后,返回页面配置参数如下,配置中 biaoZhunAction 为逻辑名称,具体业务处理 Action 类的实现指向通过 spring 配置文件实现^[5]。

```
<struts>
  <package name="hqbz" extends="struts-default">
    <action name="listBiaoZhun" class="biaoZhunAction" method="list">
      <result name="success">/common/bzcx/list_biaozhunxx.jsp</result>
    </action>
  </package>
</struts>
```

applicationContext.xml 中的配置实现业务逻辑转换,下面以标准信息相关业务处理为例,配置会话以实现 hibernate 功能整合,配置标准信息 Dao 用于操作标准信息数据库表,此处通过参数配置实现标准信息类的注入,配置标准信息 Service 用于标准信息业务逻辑处理的调用,配置标准信息 Action 用于具体指向标准信息业务处理实现类,参数配置如下^[7]。

```
<beans>
  <bean id="sessionFactory"
    class="org.springframework.orm.hibernate3.annotation.AnnotationSessionFactoryBean">
    <property name="configLocation">
      <value>/WEB-INF/classes/hibernate.cfg.xml</value>
    </property>
  </bean>
  <bean name="biaoZhunDao" class="kys.zts.hqbz.dao.impl.GenericDaoImpl">
    <constructor-arg><value>kys.zts.hqbz.bean.BiaoZhunXXBean</value></constructor-arg>
    <property name="sessionFactory"><ref bean="sessionFactory"></ref></property>
  </bean>
  <bean name="biaoZhunService" class="kys.zts.hqbz.service.impl.GenericServiceImpl">
    <property name="genericDao"><ref bean="biaoZhunDao"></ref></property>
  </bean>
  <bean name="biaoZhunAction" class="kys.zts.hqbz.action.BiaoZhunXXAction" singleton="false">
    <property name="genericService"><ref bean="biaoZhunService"></ref></property>
```

```
aoZhunService"></ref></property>
  </bean>
</beans>
```

3 结束语

文中对基于 SSH 框架的 Web 应用中常见的分页处理功能进行分析,设计并实现了通用页面信息存储类和通用业务处理 Action 类,同时在有效整合 struts、spring、hibernate 的基础上,设计出一种通用分页方法模型,最后通过标准信息分页处理实例详细阐述了该模型的实现过程。

该分页方法通过配置文件中的参数化配置注入业务 Bean 类型,提高了代码的复用性,降低了编程的复杂度;应用 struts2 实现前台分页信息的展现,spring 实现具体分页处理逻辑转换,hibernate 实现对符合分页条件数据库记录的查询,通过逻辑分层使得分页处理过程更加简单清晰、易于实现。

参考文献:

- [1] 庞志赞. 基于 J2EE 平台的 WEB 教学系统集成研究[J]. 计算机技术与发展, 2010, 20(12): 107-110.
- [2] 田秀彦, 李忠, 罗士美, 等. 基于 Struts+Hibernate+Spring 的整合架构及其在 Web 开发中的应用[J]. 计算机与现代化, 2008(11): 95-98.
- [3] 刘斌, 王最. Struts, Spring, Hibernate 框架在 OA 开发中的应用[J]. 计算机技术与发展, 2010, 20(1): 151-154.
- [4] 李发英, 朱海滨. 基于 Struts+Hibernate 的 Web 应用的设计与实现[J]. 计算机技术与发展, 2009, 19(4): 91-93.
- [5] Cavaness C. Programming Jakarta Struts [M]. Gravenstein Highway North, Sebastopol, CA: O'Reilly & Associates, Inc, 2002.
- [6] 裴承丹, 王春梅. 基于 Struts 2/Spring/Hibernate 框架的名片管理系统的设计与实现[J]. 软件导刊, 2008, 7(7): 139-140.
- [7] Jotmsom R. Spring framework reference documentation [EB/OL]. 2004. <http://www.springframework.org/documentation>.
- [8] 石扬, 张燕平. 基于 Struts + Spring + Hibernate 的 Web-MIS 开发研究[J]. 计算机技术与发展, 2007, 17(1): 46-48.
- [9] 周铁成, 陈忠文, 熊晖. 基于 Struts2 + JPA + Spring2 架构的 IT 服务台系统的设计与实现[J]. 科学技术与工程, 2009, 5(9): 1290-1293.
- [10] Hibernate Reference Documentation [EB/OL]. 2005-04. <http://www.hibernate.org/hib-docs/v3/reference/en/html/>.
- [11] 李红梅, 王坚, 李巍, 等. 基于 Spring+Hibernate 的开发方法及实例[J]. 吉林大学学报, 2008, 26(6): 593-598.
- [12] 余腊生, 任炬. 基于 SSH 三层架构的 OJ 系统研究与设计[J]. 电脑知识与技术, 2008, 26(3): 89-93.