

序号: 24

学号: 16430121



# 常州大学

## 实 验 报 告

课 程 名 称: Java EE 程序设计与应用开发

实 验 题 目: Struts、Hibernate 和 Spring 的整合

实 验 次 数: 第 六 次实验      实验时间: 2018 年 5 月 7 日

学 生 姓 名: 林锦雄

学      院: 信息数理 专 业 班 级: 计算机 162

指 导 教 师: 陆洁茹 评 阅 成 绩:

## 一、实验目的：

1. 掌握 Struts, Spring, Hibernate 三种框架整合的方法，理解三种框架协调一致的编程原理。
2. 要求：使用 Struts, Spring, Hibernate 三种框架整合的方法实现一个 Web 应用的查询功能和添加功能。

## 二、实验仪器、设备

### 1、硬件环境

PC 微机；2G 以上内存；VGA 显示格式。

### 2、软件环境

Windows XP 以上操作系统，JDK，Tomcat 服务器等。

## 三、实验内容与要求

在数据库中建立表格 T\_BOOK(BOOKID,BOOKNAME,BOOKPRICE)，插入一些记录。如图 1 所示。

```
MariaDB [(none)]> use books;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [books]> select * from T_BOOK;
+-----+-----+-----+
| BOOKID | BOOKNAME | BOOKPRICE |
+-----+-----+-----+
| 10001 | 童话故事 | 32.5 |
| 10002 | 安徒生童话 | 23.8 |
| 10003 | 格林童话 | 15.8 |
| 10004 | 希腊神话全集 | 159.99 |
| 10005 | 数据结构 | 43.5 |
| 10006 | 编译原理 | 57.9 |
| 10007 | BigBen | 32.3 |
+-----+-----+-----+
7 rows in set (0.001 sec)

MariaDB [books]> 
```

图 1. books 数据库中的 T\_BOOK 表

1. 编写一个模糊查询图书的应用，输入图书名称的模糊资料，显示查询的图书 ID、名称和价格。要求使用 Struts+Spring+Hibernate 完成。

index.jsp 是程序的起始页面（即欢迎页面），在这个页面，上面是“实验 6”字样，下面是一个表格，包含本次实验的两道题目的菜单信息，每一行为一道题目，每行的最后面为进入该题目的按钮，点击对应地按钮即可进入对应的实验内容。如图 2 所示。

```

1. <!-- index.jsp -->
2. <h2>实验 6</h2><hr>
3. <table border="2">
4.   <tr>
5.     <th></th>
6.     <th>题目</th>
7.     <th>操作</th>
8.   </tr>
9.   <tr>
10.    <form action="WEB-INF/JSP/queryBook.jsp" method="post">
11.      <td>6.1</td>
12.      <td>模糊查询图书</td>
13.      <td><input type="submit" value="进入"></td>
14.    </form>
15.  </tr>
16.  <tr>
17.    <form action="deleteBook.action" method="post">
18.      <td>6.2</td>
19.      <td>全部图书资料</td>
20.      <td><input type="submit" value="进入"></td>
21.    </form>
22.  </tr>
23. </table>

```



图 2. Index.jsp 页面

queryBook.jsp 为模糊查询图书的表单页面，用户通过输入图书名称的模糊资料，即图书关键字，把模糊资料通过表单提交到后台处理，因为使用了 struts2 框架，所以是提交到相应的 Action，最后 Action 把处理好的结果传到指定页面。如图 3 所示。

```

1. <!-- queryBook.jsp -->
2. <h2>欢迎使用图书查询系统!</h2><hr>
3. <form action="queryByKeyword.action" method="post">
4.   请输入查询图书关键字: <input type="text" name="keyWord"><br><br>
5.   <input type="submit" value="查询">
6. </form>

```

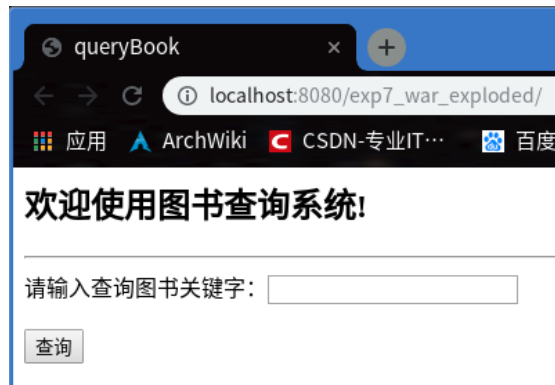


图 3. queryBook.jsp 初始页面

queryBookByKeywordAction.java 是根据图书名称关键字查询图书资料的 Action，本项目使用 Struts2 框架，所有该 queryBookByKeywordAction.java 不需要继承 Action 类，只需要通过 struts.xml 配置文件进行 action 配置即可。

```
1. //queryBookByKeywordAction
2. public class queryBookByKeywordAction {
3.     private String keyWord;
4.     private BookDao bookDao;
5.     public String execute() {
6.         ArrayList<TBook> list = bookDao.queryBookByKeyword(keyWord);
7.         Map session = ActionContext.getContext().getSession();
8.         session.put("list",list);
9.         return "success";
10.    }
11. }
```

BookDao 中的 queryBookByKeyword 方法是根据传进来的 bookName 关键字来查询数据库并返回匹配的图书资料。该 BookDao 类有一个 SessionFactory 对象，通过 spring 框架进行动态注入，然后 queryBookByKeyword 方法通过该对象打开一个 session 对象，然后生产一个 query 对象进行 HQL 查询，最后把查询结果返回。BookDao 类访问数据库整合了 hibernate 框架，可以使用方便的 HQL 查询方法。

```
1. //BookDao 中的 queryBookByKeyword
2. public class BookDao {
3.     private SessionFactory sessionFactory;
4.     public ArrayList queryBookByKeyword(String bookName) {
5.         Session session = sessionFactory.openSession();
6.         String hql = "from TBook where bookname like :name";
7.         Query query = session.createQuery(hql);
8.         query.setParameter("name","%" + bookName + "%");
9.         ArrayList<TBook> books = (ArrayList<TBook>) query.list();
10.        session.close();
11.        return books;
12.    }
13. }
```

web.xml 注册了 struts2 框架和 spring 框架的 listener 和 applicationContext.xml 配置文件的路径信息。Listener 可以截获 struts2 生产 Action 时的信息,并在 spring 框架中生产指定的 Action,这样就整合 struts2 框架和 spring 框架,让 Action 类的装配通过 spring 配置文件生成。

```
1. <!-- web.xml -->
2. <listener>
3.     <listener-class>org.springframework.web.context.ContextLoaderListener</listen
er-class>
4. </listener>
5.
6. <context-param>
7.     <param-name>contextConfigLocation</param-name>
8.     <param-value>classpath*:applicationContext.xml</param-value>
9. </context-param>
10.
11. <filter>
12.     <filter-name>struts2</filter-name>
13.     <filter-class>org.apache.struts2.dispatcher.filter.StrutsPrepareAndExecuteF
ilter</filter-class>
14. </filter>
15. <filter-mapping>
16.     <filter-name>struts2</filter-name>
17.     <url-pattern>/*</url-pattern>
18. </filter-mapping>
```

struts.xml 是 struts2 的配置文件,用来注册 action 信息。在该配置文件中,因为整合了 struts2 和 spring 框架,让 queryBookByKeyWordAction 类的装配交由 spring 框架来完成。在 struts2 生产 queryBookByKeyWordAction 类之前被在 web.xml 中的 listener 截获,然后命令其在 spring 框架中生产指定的 queryBookByKeyWordAction 类。

```
1. <!-- struts.xml -->
2. <struts>
3.     <package name="struts2" extends="struts-default">
4.         <action name="queryByKeyword" class="queryBook">
5.             <result name="success">/WEB-INF/JSP/showBook.jsp</result>
6.         </action>
7.     </package>
8. </struts>
```

TBook.hbm.xml 是 hibernate 框架的映射配置文件,把 Table.TBook 类中的属性映射到 books 数据库中的 T\_BOOK 表,此处省略 TBook 类,TBook 类中的属性和 T\_BOOK 表中的字段一一对应,对应关系如 TBook.hbm.xml 映射文件所示。所有对 TBook 类生产的对象的操作都可以映射到 books 数据库中的 T\_BOOK 表中,即可以通过操作该类生产的对象来操作数据库中的数据。

```

1. <!-- TBook.hbm.xml -->
2. <hibernate-mapping>
3. <class name="Table.TBook" table="T_BOOK" schema="books">
4.     <id name="bookid" column="BOOKID"/>
5.     <property name="bookname" column="BOOKNAME"/>
6.     <property name="bookprice" column="BOOKPRICE"/>
7. </class>
8. </hibernate-mapping>

```

hibernate.cfg.xml 是 hibernate 框架的主配置文件。该项目把 spring 框架和 hibernate 框架进行了整合，所有已经把 hibernate 配置文件中的对数据库的配置信息转交给了 spring 框架的配置文件。

```

1. <!-- hibernate.cfg.xml -->
2. <hibernate-configuration>
3.     <session-factory>
4.         <mapping class="Table.TBook"/>
5.         <mapping resource="Table/TBook.hbm.xml"/>
6.     </session-factory>
7. </hibernate-configuration>

```

applicationContext.xml 是 spring 框架的配置文件。该项目整合了 Struts 框架、Spring 框架和 Hibernate 框架，类的对象装配都交由 spring 框架完成，例如 DAO 中的 BookDao 类在 spring 框架中装配一个对象为 bookDao，还有 struts2 框架中的 queryBookByKeyWordAction 也由 spring 框架装配。hibernate 框架中和数据库的配置信息也整合到了该 spring 框架中。整合这三个框架，大大地降低了程序地耦合度，类与类之间的耦合度大大降低，不再互相依赖。详细的整合配置信息如下配置文件所示。

```

1. <!-- applicationContext.xml -->
2. <bean class="DAO.BookDao" name="bookDao">
3.     <property name="sessionFactory" ref="sessionFactory"></property>
4. </bean>
5.
6. <bean class="Action.queryBookByKeyWordAction" name="queryBook">
7.     <property name="bookDao" ref="bookDao"></property>
8. </bean>
9.
10. <bean class="org.springframework.orm.hibernate5.LocalSessionFactoryBean" name="
    sessionFactory">
11.     <property name="dataSource" ref="dataSource"></property>
12.     <property name="mappingResources" value="hibernate.cfg.xml"></property>
13.     <property name="mappingLocations" value="classpath:Table/TBook.hbm.xml"></p
    roperty>
14. </bean>
15.
16. <bean class="org.apache.commons.dbcp2.BasicDataSource" name="dataSource">
17.     <property name="driverClassName" value="com.mysql.jdbc.Driver"></property>
18.     <property name="url" value="jdbc:mysql://localhost:3306/books?characterEnco
    ding=UTF-8"></property>

```

```

19.     <property name="username" value="root"/>
20.     <property name="password" value="bingjie123"/>
21. </bean>

```

showBook.jsp 是查询结果页面，通过 Action 中保存在 session 中的图书资料来显示在页面上。设置一个查询关键字“童话”，如图 4 所示。根据该关键字查询数据库得到的结果，如图 5 所示。

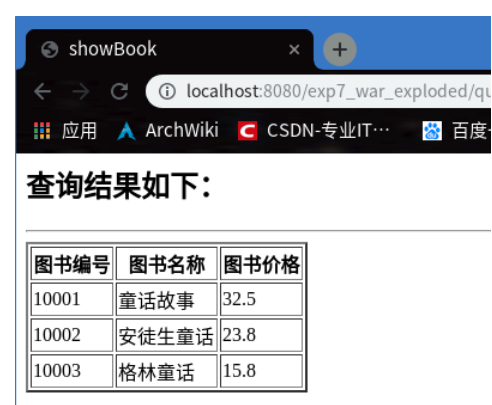
```

1. <!-- showBook.jsp -->
2. <h2>查询结果如下:</h2><hr>
3. <table border="2">
4.     <tr>
5.         <th>图书编号</th>
6.         <th>图书名称</th>
7.         <th>图书价格</th>
8.     </tr>
9.     <%
10.         Map se = ActionContext.getContext().getSession();
11.         ArrayList<TBook> books = (ArrayList<TBook>) se.get("list");
12.         for(int i = 0;i < books.size();i++) {
13.             %>
14.             <tr>
15.                 <td><%=books.get(i).getBookid()%></td>
16.                 <td><%=books.get(i).getBookname()%></td>
17.                 <td><%=books.get(i).getBookprice()%></td>
18.             </tr>
19.             <%
20.             }
21.             %>
22. </table>

```



图 4. 查询关键字图



5. 根据关键字的查询结果

2. 实现图书记录的删除功能，首先显示全部图书的资料，通过每一种图书后的“删除”链接，可以删除该图书记录。要求使用 Struts+Spring+Hibernate 完成。

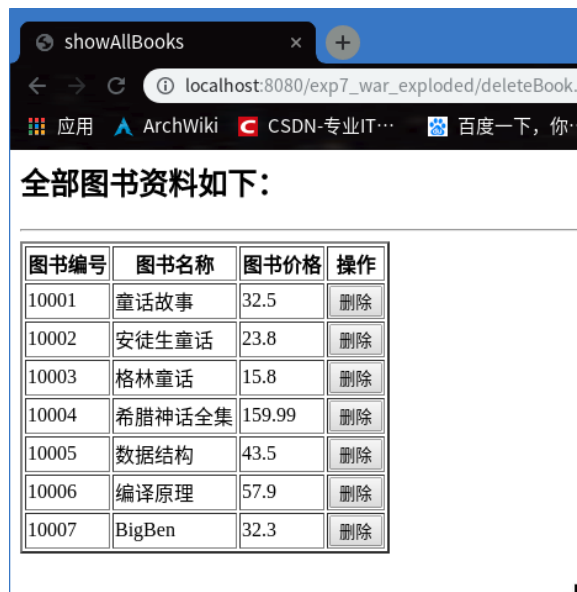


图 6. 显示全部图书资料

`deleteBookAction.java` 是根据传入的 `bookID` 来删除数据库中的指定的图书资料，并把修改后的数据库中的剩余的全部图书资料返回。在这个页面初次进入时由于没有需要删除的图书，所以直接返回数据库中全部的图书信息。

该 Action 也是由 struts2 框架交由 spring 框架装配。

```

1. //deleteBookAction
2. public class deleteBookAction {
3.     private String bookID = "";
4.     private BookDao bookDao;
5.     public String execute() {
6.         ArrayList<TBook> list = bookDao.deleteBookByID(bookID);
7.         Map session = ActionContext.getContext().getSession();
8.         session.put("list",list);
9.         return "success";
10.    }
11. }

```

`showAllBooks.jsp` 是显示查询数据库中全部图书的页面，每个图书资料后面有一个删除按钮，点击删除按钮可以删除数据库中的该图书然后根据数据库更新显示页面。

```

1. <!-- showAllBooks.jsp -->
2. <h2>全部图书资料如下:</h2><hr>
3. <table border="2">
4.     <tr>
5.         <th>图书编号</th>
6.         <th>图书名称</th>
7.         <th>图书价格</th>
8.         <th>操作</th>
9.     </tr>

```



```

10.      <%
11.          Map se = ActionContext.getContext().getSession();
12.          ArrayList<TBook> books = (ArrayList<TBook>) se.get("list");
13.          for(int i = 0;i < books.size();i++) {
14.              %>
15.              <tr>
16.                  <form action="deleteBook.action" method="post">
17.                      <input type="hidden" name="bookID" value="<%=books.get(i).getBookid()%>">
18.                      <td><%=books.get(i).getBookid()%></td>
19.                      <td><%=books.get(i).getBookname()%></td>
20.                      <td><%=books.get(i).getBookprice()%></td>
21.                      <td><input type="submit" value="删除"></td>
22.                  </form>
23.              </tr>
24.              <%
25.                  }
26.              %>
27. </table>

```

struts.xml 是 struts2 的配置文件，用来注册 action 信息。在该配置文件中，因为整合了 struts2 和 spring 框架，让 deleteBookAction 类的装配交由 spring 框架来完成。在 struts2 生产 deleteBookAction 类之前被在 web.xml 中的 listener 截获，然后命令其在 spring 框架中生产指定的 deleteBookAction 类。

在本题，struts.xml 更新一个 action 配置信息，deleteBookAction。

```

1. <!-- struts.xml -->
2. <struts>
3. <package name="struts2" extends="struts-default">
4.     <action name="deleteBook" class="deleteBook">
5.         <result name="success">/WEB-INF/JSP/showAllBooks.jsp</result>
6.     </action>
7. </package>
8. </struts>

```

applicationContext.xml 是 spring 框架的配置文件。该项目整合了 Struts 框架、Spring 框架和 Hibernate 框架，类的对象装配都交由 spring 框架完成，例如 DAO 中的 BookDao 类在 spring 框架中装配一个对象为 bookDao，还有 struts2 框架中的 queryBookByKeyWordAction 也由 spring 框架装配。hibernate 框架中和数据库的配置信息也整合到了该 spring 框架中。整合这三个框架，大大地降低了程序地耦合度，类与类之间的耦合度大大降低，不再互相依赖。详细的整合配置信息如下配置文件所示。

这里，在 applicationContext.xml 新增 deleteBookAction 信息。

```

22. <!-- applicationContext.xml -->
23. <bean class="Action.deleteBookAction" name="deleteBook">
24.     <property name="bookDao" ref="bookDao"></property>
25. </bean>

```

BookDao 中的 deleteBookByID 方法是根据传进来的 bookID 删除数据库中的匹配图书资料并返回数据库中剩余的图书资料。该 BookDao 类有一个 SessionFactory 对象,通过 spring 框架进行动态注入,然后 queryBookByKeyWord 方法通过该对象打开一个 session 对象,然后生产一个 query 对象进行 HQL 查询,最后把查询结果返回。BookDao 类访问数据库整合了 hibernate 框架,可以使用方便的 HQL 查询方法。

把全部图书资料中的最后一本书,编号为 10007 的 BigBen 图书删除,如图 7 所示。

删除 10007 号图书后的数据库 T\_BOOK 表,如图 8 所示。

```
1. //BookDao 的 deleteBookByID
2. public class BookDao {
3.     private SessionFactory sessionFactory;
4.     public ArrayList deleteBookByID(String bookID) {
5.         Session session = sessionFactory.openSession();
6.         TBook book = session.get(TBook.class,bookID);
7.         if(book != null) {
8.             Transaction ts = session.beginTransaction();
9.             session.delete(book);
10.            ts.commit();
11.        }
12.        String hql = "from TBook";
13.        Query query = session.createQuery(hql);
14.        ArrayList<TBook> books = (ArrayList<TBook>) query.list();
15.        session.close();
16.        return books;
17.    }
18. }
```



图 7. 删除图书编号 10007 的图书

```

MariaDB [(none)]> use books;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [books]> select * from T_BOOK;
+-----+-----+-----+
| BOOKID | BOOKNAME      | BOOKPRICE |
+-----+-----+-----+
| 10001  | 童话故事      | 32.5      |
| 10002  | 安徒生童话    | 23.8      |
| 10003  | 格林童话      | 15.8      |
| 10004  | 希腊神话全集  | 159.99    |
| 10005  | 数据结构      | 43.5      |
| 10006  | 编译原理      | 57.9      |
+-----+-----+-----+
6 rows in set (0.001 sec)

MariaDB [books]>

```

图 8. 删除指定图书后的数据库

## 四、实验心得

本次实验，我掌握 Struts, Spring, Hibernate 三种框架整合的方法，理解三种框架协调一致的编程原理。使用 Struts, Spring, Hibernate 三种框架整合的方法实现一个 Web 应用的查询功能和添加功能。加深了我对三个框架的理解和使用，以及三个框架整合的方法，使用这三个框架来进行编程，大大地方便了编程，使得网页开发变得更加有条理。

Struts2 是一个基于 MVC 设计模式的 Web 应用框架，它本质上相当于一个 servlet，在 MVC 设计模式中，Struts2 作为控制器(Controller)来建立模型与视图的数据交互。Struts 2 是 Struts 的下一代产品，是在 struts 1 和 WebWork 的技术基础上进行了合并的全新的 Struts 2 框架。其全新的 Struts 2 的体系结构与 Struts 1 的体系结构差别巨大。Struts 2 以 WebWork 为核心，采用拦截器的机制来处理用户的请求，这样的设计也使得业务逻辑控制器能够与 ServletAPI 完全脱离开，所以 Struts 2 可以理解为 WebWork 的更新产品。

Spring 是一个轻量级控制反转(IoC)和面向切面(AOP)的容器框架。Spring 是一个开源框架，是为了解决企业应用程序开发复杂性而创建的。框架的主要优势之一就是其分层架构，分层架构允许您选择使用哪一个组件，同时为 J2EE 应用程序开发提供集成的框架。

Hibernate 是一个开放源代码的对象关系映射框架，它对 JDBC 进行了非常轻量级的对象封装，它将 POJO 与数据库表建立映射关系，是一个全自动的 orm 框架，hibernate 可以自动生成 SQL 语句，自动执行，使得 Java 程序员可以随心所欲的使用对象编程思维来操纵数据库。Hibernate 可以应用在任何使用 JDBC 的场合，既可以在 Java 的客户端程序使用，也可以在 Servlet/JSP 的 Web 应用中使用，最具革命意义的是，Hibernate 可以在应用 EJB 的 J2EE 架构中取代 CMP，完成数据持久化的重任。

SSH 为 struts+spring+hibernate 的一个集成框架，是目前较流行的一种 Web 应用程序开源框架。集成 SSH 框架的系统从职责上分为四层：表示层、业务逻辑层、数据持久层和域模块层（实体层），以帮助开发人员在短期内搭建结构清晰、可复用性好、维护方便的 Web 应用程序。