这个例子使用Struts2+Spring2+Hibernate3组合来配合**Compass**2+**Lucene**2实现全文检索功能。

首先看一下工程使用到的jar包

以下是工程的文件结构：

这个工程是在MyEclipse6下开发的。数据库使用MySQL5。

新建数据库，库名：**compass**。

/\*

MySQL Data Transfer

Source Host: localhost

Source Database: **compass**

Target Host: localhost

Target Database: **compass**

Date: 2008-6-8 10:27:30

\*/

SET FOREIGN\_KEY\_CHECKS=0;

-- ----------------------------

-- Table structure for books

-- ----------------------------

CREATE TABLE `books` (

`ID` int(11) NOT NULL AUTO\_INCREMENT,

`NM` varchar(100) DEFAULT NULL,

`CNT` text,

`ISBN` char(10) DEFAULT NULL,

`PBLDT` date DEFAULT NULL,

PRIMARY KEY (`ID`)

) ENGINE=InnoDB AUTO\_INCREMENT=14 DEFAULT CHARSET=gbk;

-- ----------------------------

-- Records

-- ----------------------------

INSERT INTO `books` VALUES ('1', '**Lucene** IN Action 中文版', '这是一本关于Java的查询引擎的书。', '7121032171', '2007-01-01');

INSERT INTO `books` VALUES ('2', 'Spring 2.0 技术手册', '林信良写的一本关于Spring的书。', '7121039580', '2007-04-01');

INSERT INTO `books` VALUES ('4', '林信良学Lunece', '林信良学习Lunece的一本书', '1234567890', '2008-06-01');

INSERT INTO `books` VALUES ('9', 'Hibernate 深入浅出', '这是一本关于Hibernate的好书。', '7121006707', '2005-06-01');

INSERT INTO `books` VALUES ('10', 'Hibernate IN Action 中文版', '这是一本关于Hibernate的中文书', '1212121212', '2008-07-10');

INSERT INTO `books` VALUES ('11', 'Hibernate + Spring 中文系统', 'Hibernate + Spring 中文系统', '1234512345', '2008-06-06');

INSERT INTO `books` VALUES ('12', 'Hibernate 深入浅出2', 'Hibernate 深入浅出2', '2343454567', '2008-05-01');

INSERT INTO `books` VALUES ('13', 'Hibernate IN Action 中文版2', 'Hibernate IN Action 中文版2', '8888006707', '2009-06-01');

配置文件：

1.web.xml

<?xml version="1.0" encoding="UTF-8"?>  
<web-app version="2.4" xmlns="<http://java.sun.com/xml/ns/j2ee>"  
 xmlns:xsi="<http://www.w3.org/2001/XMLSchema-instance>"  
 xsi:schemaLocation="<http://java.sun.com/xml/ns/j2ee>   
 <http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd>">

 <display-name>**compass**</display-name>

 <context-param>  
  <param-name>log4jConfigLocation</param-name>  
  <param-value>/WEB-INF/log4j.properties</param-value>  
 </context-param>

 <context-param>  
  <param-name>contextConfigLocation</param-name>  
  <param-value>  
   /WEB-INF/applicationContext.xml,  
   /WEB-INF/serviceContext.xml,   
   /WEB-INF/dataAccessContext.xml,  
   /WEB-INF/compassContext.xml  
  </param-value>  
 </context-param>

 <filter>  
  <filter-name>**compass**</filter-name>  
  <filter-class>  
   org.apache.struts2.dispatcher.FilterDispatcher  
  </filter-class>  
 </filter>

 <filter-mapping>  
  <filter-name>**compass**</filter-name>  
  <url-pattern>/\*</url-pattern>  
 </filter-mapping>

 <filter>  
  <filter-name>encodingFilter</filter-name>  
  <filter-class>  
   org.springframework.web.filter.CharacterEncodingFilter  
  </filter-class>  
  <init-param>  
   <param-name>encoding</param-name>  
   <param-value>utf-8</param-value>  
  </init-param>  
 </filter>

 <filter-mapping>  
  <filter-name>encodingFilter</filter-name>  
  <url-pattern>/\*</url-pattern>  
 </filter-mapping>

 <listener>  
  <listener-class>  
   org.springframework.web.context.ContextLoaderListener  
  </listener-class>  
 </listener>

 <listener>  
  <listener-class>  
   org.springframework.web.util.Log4jConfigListener  
  </listener-class>  
 </listener>

 <welcome-file-list>  
  <welcome-file>index.jsp</welcome-file>  
 </welcome-file-list>

</web-app>

=========================================================================

2.log4j.properties

#Log4j  
log4j.rootLogger=error, stdout  
log4j.appender.stdout=org.apache.log4j.ConsoleAppender  
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout  
log4j.appender.stdout.layout.ConversionPattern=%d %p %c - %m%n

log4j.logger.org.**compass**=error

=========================================================================

3.jdbc.properties

#DataSource  
jdbc.driverClassName=com.mysql.jdbc.Driver  
jdbc.url=jdbc:mysql://127.0.0.1:3306/**compass**?characterEncoding=gbk  
jdbc.username=root  
jdbc.password=root  
jdbc.initialPoolSize=10  
jdbc.minPoolSize=5  
jdbc.maxPoolSize=30  
jdbc.acquireIncrement=5  
jdbc.maxIdleTime=10  
jdbc.maxStatements=0

#Hibernate  
hibernate.dialect=org.hibernate.dialect.MySQLDialect  
hibernate.show\_sql=true  
hibernate.jdbc.batch\_size=20  
hibernate.generate\_statistics=true

=========================================================================

4.paoding-dic-home.properties

paoding.dic.home=C:/paoding/dic  
paoding.dic.detector.interval=60

=========================================================================

5.struts.xml

<?xml version="1.0" encoding="UTF-8" ?>  
<!DOCTYPE struts PUBLIC  
    "-//Apache Software Foundation//DTD Struts Configuration 2.0//EN"  
    "<http://struts.apache.org/dtds/struts-2.0.dtd>">

<struts>  
    <constant name="struts.enable.DynamicMethodInvocation" value="false" />  
    <constant name="struts.devMode" value="false" />

    <package name="soutyuu" extends="struts-default">  
      
        <action name="\*book" class="com.cecltd.web.BookAction" method="{1}">  
   <result name="insert">jsp/insert.jsp</result>  
   <result name="search">jsp/search.jsp</result>  
  </action>  
          
    </package>  
</struts>

=========================================================================

6.**compass**.cfg.xml

<?xml version="1.0" encoding="UTF-8"?>  
<**compass**-core-config  
 xmlns="[http://www.**compass**-project.org/schema/core-config](http://www.compass-project.org/schema/core-config)"  
 xmlns:xsi="<http://www.w3.org/2001/XMLSchema-instance>"  
 xsi:schemaLocation="[http://www.**compass**-project.org/schema/core-config](http://www.compass-project.org/schema/core-config)  
           [http://www.**compass**-project.org/schema/**compass**-core-config-2.0.xsd](http://www.compass-project.org/schema/compass-core-config-2.0.xsd)">

 <**compass** name="default">  
  <connection>  
   <file path="C:/target" />  
  </connection>

  <searchEngine>  
   <analyzer name="MyAnalyzer" type="CustomAnalyzer"  
    analyzerClass="net.paoding.analysis.analyzer.PaodingAnalyzer">  
   </analyzer>  
  </searchEngine>  
 </**compass**>

</**compass**-core-config>

=========================================================================

7.applicationContext.xml

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="<http://www.springframework.org/schema/beans>"  
 xmlns:xsi="<http://www.w3.org/2001/XMLSchema-instance>"  
 xmlns:aop="<http://www.springframework.org/schema/aop>"  
 xmlns:tx="<http://www.springframework.org/schema/tx>"  
 xsi:schemaLocation="<http://www.springframework.org/schema/beans> <http://www.springframework.org/schema/beans/spring-beans-2.0.xsd>  
           <http://www.springframework.org/schema/aop> <http://www.springframework.org/schema/aop/spring-aop-2.0.xsd>  
           <http://www.springframework.org/schema/tx> <http://www.springframework.org/schema/tx/spring-tx-2.0.xsd>">

 <bean id="propertyConfigurer" class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer">  
  <property name="location">  
   <value>/WEB-INF/jdbc.properties</value>  
  </property>  
 </bean>

 <bean id="dataSource" class="com.mchange.v2.c3p0.ComboPooledDataSource" destroy-method="close">  
  <property name="driverClass">  
   <value>${jdbc.driverClassName}</value>  
  </property>  
  <property name="jdbcUrl">  
   <value>${jdbc.url}</value>  
  </property>  
  <property name="user">  
   <value>${jdbc.username}</value>  
  </property>  
  <property name="password">  
   <value>${jdbc.password}</value>  
  </property>  
  <property name="initialPoolSize">  
   <value>${jdbc.initialPoolSize}</value>  
  </property>  
  <property name="minPoolSize">  
   <value>${jdbc.minPoolSize}</value>  
  </property>  
  <property name="maxPoolSize">  
   <value>${jdbc.maxPoolSize}</value>  
  </property>  
  <property name="acquireIncrement">  
   <value>${jdbc.acquireIncrement}</value>  
  </property>  
  <property name="maxIdleTime">  
   <value>${jdbc.maxIdleTime}</value>  
  </property>  
  <property name="maxStatements">  
   <value>${jdbc.maxStatements}</value>  
  </property>  
 </bean>  
   
 <bean id="sessionFactory" class="org.springframework.orm.hibernate3.LocalSessionFactoryBean">  
  <property name="dataSource">  
   <ref bean="dataSource" />  
  </property>  
  <property name="mappingResources">  
   <list>  
    <value>com/cecltd/domain/Book.hbm.xml</value>  
   </list>  
  </property>  
  <property name="hibernateProperties">  
   <props>  
    <prop key="hibernate.dialect">  
     ${hibernate.dialect}  
    </prop>  
    <prop key="hibernate.show\_sql">  
     ${hibernate.show\_sql}  
    </prop>  
    <prop key="hibernate.jdbc.batch\_size">  
     ${hibernate.jdbc.batch\_size}  
    </prop>  
    <prop key="hibernate.generate\_statistics">  
     ${hibernate.generate\_statistics}  
    </prop>  
   </props>  
  </property>  
  <property name="eventListeners">  
   <map>  
    <entry key="merge">  
     <bean  
      class="org.springframework.orm.hibernate3.support.IdTransferringMergeEventListener" />  
    </entry>  
   </map>  
  </property>  
 </bean>

 <bean id="**compass**" class="org.**compass**.spring.LocalCompassBean">  
  <property name="resourceLocations">  
   <list>  
    <value>classpath:com/cecltd/domain/Book.cmd.xml</value>  
    <value>classpath:com/cecltd/domain/Book.cpm.xml</value>  
   </list>  
  </property>  
  <property name="configLocation">  
   <value>/WEB-INF/**compass**.cfg.xml</value>  
  </property>  
  <property name="transactionManager">  
   <ref local="transactionManager" />  
  </property>  
 </bean>

 <bean id="hibernateGpsDevice" class="org.**compass**.spring.device.hibernate.dep.SpringHibernate3GpsDevice">  
  <property name="name">  
   <value>hibernateDevice</value>  
  </property>  
  <property name="sessionFactory">  
   <ref local="sessionFactory" />  
  </property>  
 </bean>

 <bean id="compassGps" class="org.**compass**.gps.impl.SingleCompassGps" init-method="start" destroy-method="stop">  
  <property name="**compass**">  
   <ref bean="**compass**" />  
  </property>  
  <property name="gpsDevices">  
   <list>  
    <bean  
     class="org.**compass**.spring.device.SpringSyncTransactionGpsDeviceWrapper">  
     <property name="gpsDevice" ref="hibernateGpsDevice" />  
    </bean>  
   </list>  
  </property>  
 </bean>

 <bean id="transactionManager" class="org.springframework.orm.hibernate3.HibernateTransactionManager">  
  <property name="sessionFactory">  
   <ref local="sessionFactory" />  
  </property>  
 </bean>  
</beans>

=========================================================================

8.serviceContext.xml

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="<http://www.springframework.org/schema/beans>"  
 xmlns:xsi="<http://www.w3.org/2001/XMLSchema-instance>"  
 xmlns:aop="<http://www.springframework.org/schema/aop>"  
 xmlns:tx="<http://www.springframework.org/schema/tx>"  
 xsi:schemaLocation="<http://www.springframework.org/schema/beans> <http://www.springframework.org/schema/beans/spring-beans-2.0.xsd>  
           <http://www.springframework.org/schema/aop> <http://www.springframework.org/schema/aop/spring-aop-2.0.xsd>  
           <http://www.springframework.org/schema/tx> <http://www.springframework.org/schema/tx/spring-tx-2.0.xsd>">

 <bean id="bookService" class="com.cecltd.service.impl.BookServiceImpl">  
  <property name="bookDao">  
   <ref bean="bookDao" />  
  </property>  
  <property name="bookCps">  
   <ref bean="bookCps" />  
  </property>  
 </bean>

</beans>

=========================================================================

9.dataAccessContext.xml

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="<http://www.springframework.org/schema/beans>"  
 xmlns:xsi="<http://www.w3.org/2001/XMLSchema-instance>"  
 xmlns:aop="<http://www.springframework.org/schema/aop>"  
 xmlns:tx="<http://www.springframework.org/schema/tx>"  
 xsi:schemaLocation="<http://www.springframework.org/schema/beans> <http://www.springframework.org/schema/beans/spring-beans-2.0.xsd>  
           <http://www.springframework.org/schema/aop> <http://www.springframework.org/schema/aop/spring-aop-2.0.xsd>  
           <http://www.springframework.org/schema/tx> <http://www.springframework.org/schema/tx/spring-tx-2.0.xsd>">

 <bean id="bookDao" class="org.springframework.transaction.interceptor.TransactionProxyFactoryBean">  
  <property name="transactionManager">  
   <ref bean="transactionManager" />  
  </property>  
  <property name="target">  
   <ref local="bookDaoTarget" />  
  </property>  
  <property name="transactionAttributes">  
   <props>  
    <prop key="search\*">PROPAGATION\_REQUIRED,readOnly</prop>  
    <prop key="insert\*">PROPAGATION\_REQUIRED</prop>  
   </props>  
  </property>  
 </bean>

 <bean id="bookDaoTarget" class="com.cecltd.dao.jdbc.BookDaoImpl">  
  <property name="sessionFactory">  
   <ref bean="sessionFactory" />  
  </property>  
 </bean>

</beans>

=========================================================================

10.compassContext.xml

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="<http://www.springframework.org/schema/beans>"  
 xmlns:xsi="<http://www.w3.org/2001/XMLSchema-instance>"  
 xmlns:aop="<http://www.springframework.org/schema/aop>"  
 xmlns:tx="<http://www.springframework.org/schema/tx>"  
 xsi:schemaLocation="<http://www.springframework.org/schema/beans> <http://www.springframework.org/schema/beans/spring-beans-2.0.xsd>  
           <http://www.springframework.org/schema/aop> <http://www.springframework.org/schema/aop/spring-aop-2.0.xsd>  
           <http://www.springframework.org/schema/tx> <http://www.springframework.org/schema/tx/spring-tx-2.0.xsd>">

 <bean id="bookCps" class="com.cecltd.**compass**.impl.BookCpsImpl">  
  <property name="**compass**">  
   <ref bean="**compass**" />  
  </property>  
 </bean>

</beans>

=========================================================================

11.Book.hbm.xml

<?xml version="1.0"?>  
<!DOCTYPE hibernate-mapping PUBLIC  
        "-//Hibernate/Hibernate Mapping DTD 3.0//EN"  
        "<http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd>">

<hibernate-mapping>

 <class name="com.cecltd.domain.Book" table="Books">  
  <id name="id" column="ID">  
   <generator class="native" />  
  </id>  
  <property name="name" column="NM" />  
  <property name="content" column="CNT" />  
  <property name="isbn" column="ISBN" />  
  <property name="pbl\_dt" type="timestamp" column="PBLDT" />  
 </class>

</hibernate-mapping>

=========================================================================

12.Book.cpm.xml

<?xml version="1.0"?>  
<!DOCTYPE **compass**-core-mapping PUBLIC   
    "-//**Compass**/**Compass** Core Mapping DTD 2.0//EN"  
    "[http://www.**compass**-project.org/dtd/**compass**-core-mapping-2.0.dtd](http://www.compass-project.org/dtd/compass-core-mapping-2.0.dtd)">

<**compass**-core-mapping package="com.cecltd.domain">

 <class name="Book" alias="${book.book}">  
  <id name="id" />  
  <constant>  
   <meta-data>type</meta-data>  
   <meta-data-value>java</meta-data-value>  
  </constant>  
  <property name="name">  
   <meta-data>${book.name}</meta-data>  
  </property>  
  <property name="content">  
   <meta-data>${book.content}</meta-data>  
  </property>  
  <property name="isbn">  
   <meta-data>${book.isbn}</meta-data>  
  </property>  
  <property name="pbl\_dt">  
   <meta-data>${book.pbl\_dt}</meta-data>  
  </property>  
 </class>

</**compass**-core-mapping>

=========================================================================

13.Book.cmd.xml

<?xml version="1.0"?>  
<!DOCTYPE **compass**-core-meta-data PUBLIC   
    "-//**Compass**/**Compass** Core Meta Data DTD 2.0//EN"  
    "[http://www.**compass**-project.org/dtd/**compass**-core-meta-data-2.0.dtd](http://www.compass-project.org/dtd/compass-core-meta-data-2.0.dtd)">

<**compass**-core-meta-data>

 <meta-data-group id="book" displayName="Book Meta Data">

  <description>Book Meta Data</description>  
  <uri>http://**compass**/book</uri>

  <alias id="book" displayName="Book">  
   <uri>http://**compass**/book/alias/book</uri>  
   <name>book</name>  
  </alias>

  <meta-data id="name" displayName="Name">  
   <uri>http://**compass**/book/name</uri>  
   <name>name</name>  
  </meta-data>

  <meta-data id="content" displayName="Content">  
   <uri>http://**compass**/book/content</uri>  
   <name>content</name>  
  </meta-data>

  <meta-data id="isbn" displayName="Isbn">  
   <uri>http://**compass**/book/isbn</uri>  
   <name>isbn</name>  
  </meta-data>

  <meta-data id="pbl\_dt" displayName="Pbl\_dt">  
   <uri>http://**compass**/book/pbl\_dt</uri>  
   <name format="yyyy-MM-dd">pbl\_dt</name>  
  </meta-data>

 </meta-data-group>

</**compass**-core-meta-data>

=========================================================================

JSP文件：

1.index.jsp

<%@ page contentType="text/html; charset=UTF-8"%>  
<html>  
 <head>  
  <title>页面加载中，请稍后......</title>  
  <meta http-equiv="refresh" content="1;URL=/**compass**/initbook.action">  
 </head>  
</html>

2.insert.jsp

<%@ page contentType="text/html; charset=UTF-8"%>  
<%@ taglib prefix="s" uri="/struts-tags"%>  
<html>  
 <s:head theme="ajax" />  
 <body>  
  <s:form action="insertbook.action" method="post">  
   <s:textfield key="name" label="书名" size="95" />  
   <s:textarea key="content" label="摘要" rows="10" cols="73" />  
   <s:textfield key="isbn" label="ISBN" size="13" maxlength="10" />  
   <s:datetimepicker name="pbl\_dt" label="出版日"  
    displayFormat="yyyy-MM-dd" />  
   <s:submit value=" 注册 " />  
  </s:form>  
 </body>  
</html>

3.search.jsp

<%@ page contentType="text/html; charset=UTF-8"%>  
<%@ taglib prefix="s" uri="/struts-tags"%>  
<html>  
 <head>  
  <title>结果显示</title>  
 </head>  
 <body>  
  <s:form action="searchbook.action" method="post">  
   <s:textfield key="query" label="关键字" size="118" />  
   <s:submit value=" 搜索 " onclick="this.form.action='searchbook.action'" />  
   <s:submit value=" 返回 " onclick="this.form.action='gobackbook.action'" />  
   <table id="dtTable" width="800" border="1" cellpadding="0"  
    cellspacing="0" style="BORDER-COLLAPSE: collapse"  
    borderColor=#000000>  
    <thead>  
     <tr bgcolor="orange" height="30px">  
      <th width="25%">  
       书名  
      </th>  
      <th width="45%">  
       概要  
      </th>  
      <th width="15%">  
       ISBN  
      </th>  
      <th width="15%">  
       出版日  
      </th>  
     </tr>  
    </thead>  
    <tbody>  
     <s:iterator value="datalist" status="rs">  
      <tr align="center" height="26px">  
       <td>  
        <s:property value="name" />  
       </td>  
       <td>  
        <s:property value="content" />  
       </td>  
       <td>  
        <s:property value="isbn" />  
       </td>  
       <td>  
        <s:property value="pbl\_dt" />  
       </td>  
      </tr>  
     </s:iterator>  
    </tbody>  
   </table>  
  </s:form>  
 </body>  
</html>

=========================================================================

Java文件

1.Book.java

[view plaincopy to clipboardprint?](http://blog.csdn.net/kunshan_shenbin/archive/2008/06/08/2522698.aspx)

1. **package** com.cecltd.domain;
3. **import** java.io.Serializable;
4. **import** java.util.Date;
6. **public** **class** Book **implements** Serializable {
8. **private** **static** **final** **long** serialVersionUID = -5275017032598185911L;
10. **private** Integer id;
11. **private** String name;
12. **private** String content;
13. **private** String isbn;
14. **private** Date pbl\_dt;
16. **public** Integer getId() {
17. **return** id;
18. }
19. **public** **void** setId(Integer id) {
20. **this**.id = id;
21. }
22. **public** String getName() {
23. **return** name;
24. }
25. **public** **void** setName(String name) {
26. **this**.name = name;
27. }
28. **public** String getContent() {
29. **return** content;
30. }
31. **public** **void** setContent(String content) {
32. **this**.content = content;
33. }
34. **public** String getIsbn() {
35. **return** isbn;
36. }
37. **public** **void** setIsbn(String isbn) {
38. **this**.isbn = isbn;
39. }
40. **public** Date getPbl\_dt() {
41. **return** pbl\_dt;
42. }
43. **public** **void** setPbl\_dt(Date pbl\_dt) {
44. **this**.pbl\_dt = pbl\_dt;
45. }
46. }

package com.cecltd.domain;

import java.io.Serializable;

import java.util.Date;

public class Book implements Serializable {

private static final long serialVersionUID = -5275017032598185911L;

private Integer id;

private String name;

private String content;

private String isbn;

private Date pbl\_dt;

public Integer getId() {

return id;

}

public void setId(Integer id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getContent() {

return content;

}

public void setContent(String content) {

this.content = content;

}

public String getIsbn() {

return isbn;

}

public void setIsbn(String isbn) {

this.isbn = isbn;

}

public Date getPbl\_dt() {

return pbl\_dt;

}

public void setPbl\_dt(Date pbl\_dt) {

this.pbl\_dt = pbl\_dt;

}

}

2.BookAction.java

[view plaincopy to clipboardprint?](http://blog.csdn.net/kunshan_shenbin/archive/2008/06/08/2522698.aspx)

1. **package** com.cecltd.web;
3. **import** java.text.SimpleDateFormat;
4. **import** java.util.List;
6. **import** com.cecltd.domain.Book;
7. **import** com.cecltd.service.BookService;
8. **import** com.opensymphony.xwork2.ActionSupport;
10. **public** **class** BookAction **extends** ActionSupport {
12. **private** **static** **final** **long** serialVersionUID = -3245236770369400667L;
14. **public** **static** **final** String INSERT = "insert";
15. **public** **static** **final** String SEARCH = "search";
17. **private** BookService bookService;
18. **public** BookAction(BookService bookService) {
19. **this**.bookService = bookService;
20. }
22. **public** String insert() **throws** Exception {
24. Book book = **new** Book();
25. book.setName(getName());
26. book.setContent(getContent());
27. book.setIsbn(getIsbn());
28. SimpleDateFormat format = **new** SimpleDateFormat("yyyy-MM-dd");
29. book.setPbl\_dt(format.parse(getPbl\_dt()));
31. bookService.insertBook(book);
33. **return** SEARCH;
34. }
36. **public** String search() **throws** Exception {
38. List list = bookService.searchBook(getQuery());
39. setDatalist(list);
41. **return** SEARCH;
42. }
44. **public** String init() **throws** Exception {
45. **return** INSERT;
46. }
48. **public** String goback() **throws** Exception {
49. **return** INSERT;
50. }
52. **private** String name;
53. **private** String content;
54. **private** String isbn;
55. **private** String pbl\_dt;
57. **private** String query;
58. **private** List datalist;
60. **public** String getQuery() {
61. **return** query;
62. }
64. **public** **void** setQuery(String query) {
65. **this**.query = query;
66. }
68. **public** List getDatalist() {
69. **return** datalist;
70. }
72. **public** **void** setDatalist(List datalist) {
73. **this**.datalist = datalist;
74. }
76. **public** String getName() {
77. **return** name;
78. }
80. **public** **void** setName(String name) {
81. **this**.name = name;
82. }
84. **public** String getContent() {
85. **return** content;
86. }
88. **public** **void** setContent(String content) {
89. **this**.content = content;
90. }
92. **public** String getIsbn() {
93. **return** isbn;
94. }
96. **public** **void** setIsbn(String isbn) {
97. **this**.isbn = isbn;
98. }
100. **public** String getPbl\_dt() {
101. **return** pbl\_dt;
102. }
104. **public** **void** setPbl\_dt(String pbl\_dt) {
105. **this**.pbl\_dt = pbl\_dt;
106. }
107. }

package com.cecltd.web;

import java.text.SimpleDateFormat;

import java.util.List;

import com.cecltd.domain.Book;

import com.cecltd.service.BookService;

import com.opensymphony.xwork2.ActionSupport;

public class BookAction extends ActionSupport {

private static final long serialVersionUID = -3245236770369400667L;

public static final String INSERT = "insert";

public static final String SEARCH = "search";

private BookService bookService;

public BookAction(BookService bookService) {

this.bookService = bookService;

}

public String insert() throws Exception {

Book book = new Book();

book.setName(getName());

book.setContent(getContent());

book.setIsbn(getIsbn());

SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd");

book.setPbl\_dt(format.parse(getPbl\_dt()));

bookService.insertBook(book);

return SEARCH;

}

public String search() throws Exception {

List list = bookService.searchBook(getQuery());

setDatalist(list);

return SEARCH;

}

public String init() throws Exception {

return INSERT;

}

public String goback() throws Exception {

return INSERT;

}

private String name;

private String content;

private String isbn;

private String pbl\_dt;

private String query;

private List datalist;

public String getQuery() {

return query;

}

public void setQuery(String query) {

this.query = query;

}

public List getDatalist() {

return datalist;

}

public void setDatalist(List datalist) {

this.datalist = datalist;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getContent() {

return content;

}

public void setContent(String content) {

this.content = content;

}

public String getIsbn() {

return isbn;

}

public void setIsbn(String isbn) {

this.isbn = isbn;

}

public String getPbl\_dt() {

return pbl\_dt;

}

public void setPbl\_dt(String pbl\_dt) {

this.pbl\_dt = pbl\_dt;

}

}

3.BookService.java

[view plaincopy to clipboardprint?](http://blog.csdn.net/kunshan_shenbin/archive/2008/06/08/2522698.aspx)

1. **package** com.cecltd.service;
3. **import** java.util.List;
5. **import** com.cecltd.domain.Book;
7. **public** **interface** BookService {
9. **public** List searchBook(String query);
11. **public** **void** insertBook(Book book);
12. }

package com.cecltd.service;

import java.util.List;

import com.cecltd.domain.Book;

public interface BookService {

public List searchBook(String query);

public void insertBook(Book book);

}

4.BookServiceImpl.java

[view plaincopy to clipboardprint?](http://blog.csdn.net/kunshan_shenbin/archive/2008/06/08/2522698.aspx)

1. **package** com.cecltd.service.impl;
3. **import** java.util.List;
5. **import** com.cecltd.**compass**.BookCps;
6. **import** com.cecltd.dao.BookDao;
7. **import** com.cecltd.domain.Book;
8. **import** com.cecltd.service.BookService;
10. **public** **class** BookServiceImpl **implements** BookService {
12. **private** BookDao bookDao;
13. **private** BookCps bookCps;
15. **public** **void** setBookDao(BookDao bookDao) {
16. **this**.bookDao = bookDao;
17. }
19. **public** **void** setBookCps(BookCps bookCps) {
20. **this**.bookCps = bookCps;
21. }
23. **public** **void** insertBook(Book book) {
24. bookDao.insertBook(book);
25. }
27. **public** List searchBook(String query) {
28. **return** bookCps.searchBook(query);
29. }
30. }

package com.cecltd.service.impl;

import java.util.List;

import com.cecltd.**compass**.BookCps;

import com.cecltd.dao.BookDao;

import com.cecltd.domain.Book;

import com.cecltd.service.BookService;

public class BookServiceImpl implements BookService {

private BookDao bookDao;

private BookCps bookCps;

public void setBookDao(BookDao bookDao) {

this.bookDao = bookDao;

}

public void setBookCps(BookCps bookCps) {

this.bookCps = bookCps;

}

public void insertBook(Book book) {

bookDao.insertBook(book);

}

public List searchBook(String query) {

return bookCps.searchBook(query);

}

}

5.BookDao.java

[view plaincopy to clipboardprint?](http://blog.csdn.net/kunshan_shenbin/archive/2008/06/08/2522698.aspx)

1. **package** com.cecltd.dao;
3. **import** com.cecltd.domain.Book;
5. **public** **interface** BookDao {
7. **public** **void** insertBook(Book book);
8. }

package com.cecltd.dao;

import com.cecltd.domain.Book;

public interface BookDao {

public void insertBook(Book book);

}

6.BookDaoImpl.java

[view plaincopy to clipboardprint?](http://blog.csdn.net/kunshan_shenbin/archive/2008/06/08/2522698.aspx)

1. **package** com.cecltd.dao.jdbc;
3. **import** org.springframework.orm.hibernate3.HibernateTemplate;
5. **import** com.cecltd.dao.BookDao;
6. **import** com.cecltd.domain.Book;
8. **public** **class** BookDaoImpl **extends** HibernateTemplate **implements** BookDao {
10. **public** **void** insertBook(Book book) {
12. save(book);
13. }
14. }

package com.cecltd.dao.jdbc;

import org.springframework.orm.hibernate3.HibernateTemplate;

import com.cecltd.dao.BookDao;

import com.cecltd.domain.Book;

public class BookDaoImpl extends HibernateTemplate implements BookDao {

public void insertBook(Book book) {

save(book);

}

}

7.BookCps.java

[view plaincopy to clipboardprint?](http://blog.csdn.net/kunshan_shenbin/archive/2008/06/08/2522698.aspx)

1. **package** com.cecltd.**compass**;
3. **import** java.util.List;
5. **public** **interface** BookCps {
7. **public** List searchBook(String query);
8. }

package com.cecltd.**compass**;

import java.util.List;

public interface BookCps {

public List searchBook(String query);

}

8.BookCpsImpl.java

[view plaincopy to clipboardprint?](http://blog.csdn.net/kunshan_shenbin/archive/2008/06/08/2522698.aspx)

1. **package** com.cecltd.**compass**.impl;
3. **import** java.util.ArrayList;
4. **import** java.util.List;
6. **import** org.**compass**.core.CompassHits;
7. **import** org.**compass**.core.CompassQuery;
8. **import** org.**compass**.core.CompassTemplate;
9. **import** org.**compass**.core.CompassTransaction;
11. **import** com.cecltd.**compass**.BookCps;
12. **import** com.cecltd.domain.Book;
14. **public** **class** BookCpsImpl **extends** CompassTemplate **implements** BookCps {
16. **public** List searchBook(String query) {
18. CompassTransaction tx = getCompass().openSession().beginTransaction();
19. //CompassHits hits = find("name:"+ query);
20. CompassHits hits = getCompass().openSession().queryBuilder().queryString("content:"+ query).toQuery().
21. addSort("isbn", CompassQuery.SortPropertyType.STRING, CompassQuery.SortDirection.REVERSE).hits();
23. List<BOOK></BOOK> list = **new** ArrayList<BOOK></BOOK>();
24. **int** len = hits.getLength();
25. **for** (**int** i = 0; i < len; i++) {
26. list.add((Book)hits.data(i));
27. }
28. tx.commit();
30. **return** list;
31. }
32. }

package com.cecltd.**compass**.impl;

import java.util.ArrayList;

import java.util.List;

import org.**compass**.core.CompassHits;

import org.**compass**.core.CompassQuery;

import org.**compass**.core.CompassTemplate;

import org.**compass**.core.CompassTransaction;

import com.cecltd.**compass**.BookCps;

import com.cecltd.domain.Book;

public class BookCpsImpl extends CompassTemplate implements BookCps {

public List searchBook(String query) {

CompassTransaction tx = getCompass().openSession().beginTransaction();

//CompassHits hits = find("name:"+ query);

CompassHits hits = getCompass().openSession().queryBuilder().queryString("content:"+ query).toQuery().

addSort("isbn", CompassQuery.SortPropertyType.STRING, CompassQuery.SortDirection.REVERSE).hits();

List list = new ArrayList();

int len = hits.getLength();

for (int i = 0; i < len; i++) {

list.add((Book)hits.data(i));

}

tx.commit();

return list;

}

}

注意，因为该系统中使用到了庖丁解牛中文分词方法，需要自行下载安装并放置在配置文件中所指明的地方。

庖丁解牛：<http://groups.google.com/group/paoding/browse_thread/thread/9771c8d495786fee>

另外还有IKAnalyzer：<http://groups-beta.google.com/group/SegWord/web/segword>