

基于 Struts2 框架图书管理系统设计与实现

目 录

1、设计数据库	2
2、新建 web Project 项目	3
4、引入 Struts2 包.....	3
5、设计登陆界面	5
6、建立不同的包	5
7、编写 pojo 对象.....	5
8、导入 SQLHelper.java，src 中添加 dbinfo.properties	6
9、编写用户登陆	6
10、编写 struts.xml	7
11、编写 LoginAction	7
13、编写 showbook.jsp 页面	11
14、分页 GetPageAction	12
15、编写 bookAction.....	13
16、编写 modifyUI.jsp 页面.....	15

基于 Struts2 框架图书管理系统设计与实现

1、设计数据库

```
create database Shopping
on
( name='Shopping_data',
  filename='d:\Shopping_data.mdf',
  size=10mb
)
log on
( name='Shopping_log',
  filename='d:\Shopping_log.ldf',
  size=10mb
)
go
use Shopping
go
create table users --创建用户登录表
(
  id int identity(1,1) primary key, --标签编号 为自增长字段，增长步长为 1，主码
  username nvarchar(20) not null, --用户姓名 可变字符串类型，长度 20
  [password] nvarchar(20) not null, --密码 可变字符串类型，长度 20
  email nvarchar(50) not null,
  tel nvarchar(20) not null,
  grade int default 1 not null
)

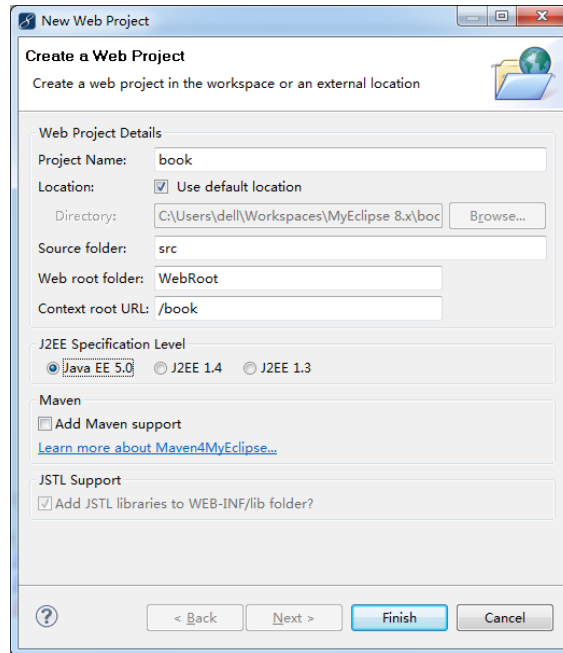
insert into users values('wl','123','wl@126.com','1303030333',2)
insert into users values('admin','123','admin@126.com','346789',2)

create table book
(
  id int identity(1,1) primary key, --标签编号 为自增长字段，增长步长为 1，主码
  bookname nvarchar(50) not null, --书名 可变字符串类型，长度 20
  author nvarchar(50) not null,
  publishHouse nvarchar(200) not null,
  price numeric(10,3) not null,
  nums numeric(10,3) default 1000
)
insert into book values('jsp 应用开发详解','刘晓华','电子工业出版社',59.80,500)
insert into book values('Java 项目开发案例全程实录','李钟尉,陈丹丹','清华大学出版社',69.80,200)
insert into book values('ASP.NET 项目开发案例全程实录','郑齐心,房大伟,刘云峰','清华大学出版社',74.00,456)
insert into book values('jquery 技术内幕：深入解析 jquery 架构设计与实现原理','高云','机械工业出版社',99.00,200)
```

```
insert into book values('轻量级 java ee 企业应用实战','李刚','电子工业出版社',99.00,100)
insert into book values('Struts+Spring+Hibernate 框架及应用开发','王建国,王建英','清华大学出版社',69,2000)
insert into book values('Java 编程思想(第 4 版)','Bruce Eckel','机械工业出版社',108,3000)
insert into book values('Java 从入门到精通','明日科技','清华大学出版社',59.80,100)
insert into book values('java 核心技术基础知识(原书第 9 版)','Cay S.Horstmann;Gary Cornell','机械工业出版社',119,300)
insert into book values('C#从入门到精通(第 3 版)','明日科技','清华大学出版社',69.80,200)
```

2、新建 web Project 项目

启动 MyEclipse，新建 web Project 项目，取名 book。注意选取 javaEE5.0



3、导入 sql 驱动包

为了能够使用数据库，在 book->src->WebRoot->lib 下导入 sql 驱动包，复制粘贴即可。项目会自动加载驱动。

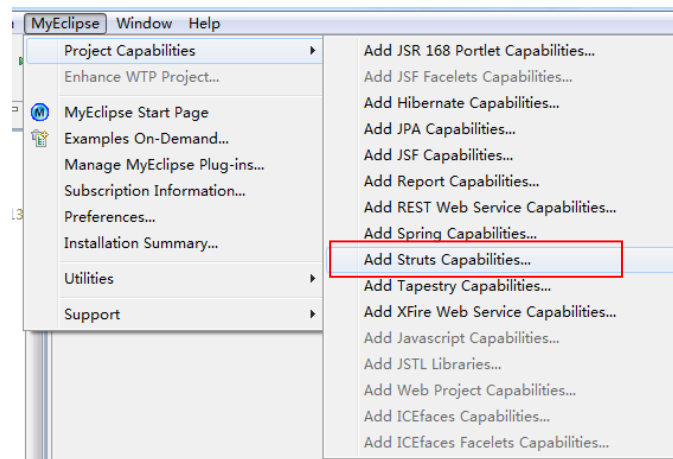


sqljdbc4.jar

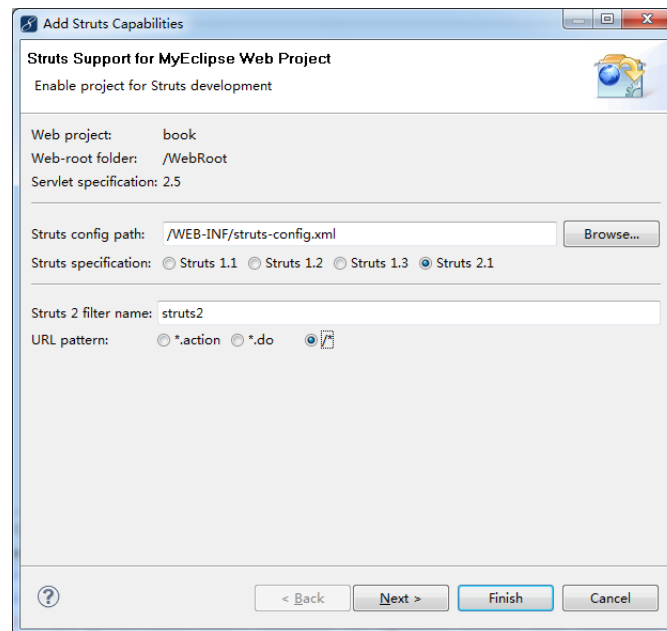
4、引入 Struts2 包

为了能够使用 Struts2，需要引入 Struts2 包。

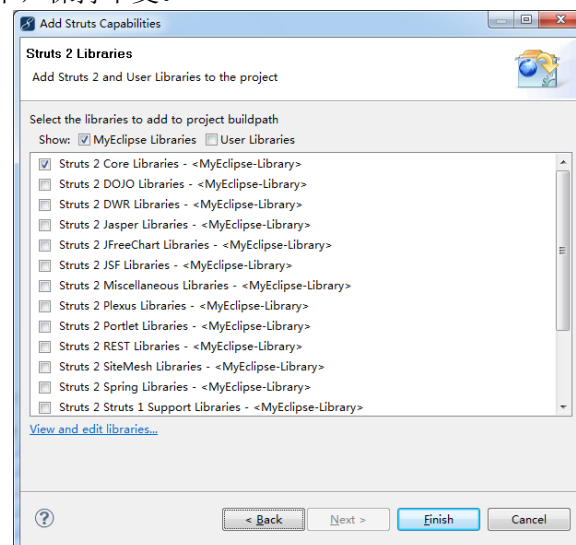
(1) MyEclipse->Project Capabilities->Add Struts Capabilities...



(2) Struts specification 中选择 Struts 2.1, Struts 2 filter name 中选择 /*。



(3) Add Struts Capabilities 下, 保持不变。



(4) Struts2 包自动导入成功!

5、设计登陆界面

在 WebRoot 下新建 login.jsp，设计登陆界面。

Struts图书管理系统

用户登陆

登陆名:	<input type="text"/>
密 码:	<input type="password"/>
<input type="submit" value="登陆"/> <input type="reset" value="清空"/>	

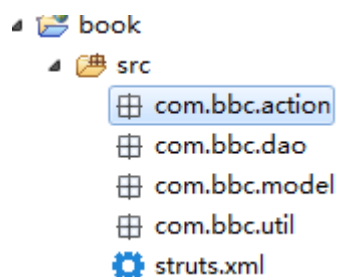
代码如下:

```
<body>
<center><h1>Struts图书管理系统</h1></center>
<center><h1>用户登陆</h1></center>
<form action="login.action" method="post">
<center>
<table border="1" width="300px" bordercolor="green" cellspacing="0">
<tr><td>登陆名: </td><td><input type="text" name="user.username"/></td></tr>
<tr><td>密 码: </td><td><input type="password" name=" user.password"/></td></tr>
<tr><td colspan="2"><input type="submit" value="登陆"><input type="reset" value="清空
"/></td></tr>
</table>
</center>
</form>
</body>
```

注意: struts2 中传递参数, 常用的是使用域模型的方式。因此要把该页面中要提交的文本框封装成类。必须使用: name="user.username" name=" user.password" 而不是 name="username" name=" password"

6、建立不同的包

在项目下建立不同的包, 以便分类管理



7、编写 pojo 对象

在 com.bbc.model 中编写 pojo 对象。

(1) user 对象

```
private Integer id;
```

```

private String username;
private String password;
private String email;
private String tel;
private Integer grade;

```

set 和 get 方法省略

(2) book 对象

```

private Integer id;
private String bookname;
private String author;
private String publishHouse;
private Float price;
private Integer nums;

```

set 和 get 方法省略

8、导入 SQLHelper.java，src 中添加 dbinfo.properties

在 com.bbc.util 包中导入 SQLHelper.java。适当修改 executeQuery2 中的内容。在 src 中添加 dbinfo.properties。

dbinfo.properties	
name	value
url	jdbc:sqlserver://localhost:1433;DatabaseName=Shopping
driver	com.microsoft.sqlserver.jdbc.SQLServerDriver
user	sa
password	123456

9、编写用户登陆

在 com.bbc.dao 中编写业务逻辑类处理用户登陆。

```

package com.bbc.dao;
import java.sql.ResultSet;
import java.sql.SQLException;
import com.bbc.model.User;
import com.bbc.util.SQLHelper;
public class UserDAO {
// 根据用户名和密码取出后台数据库信息
public boolean checkUser(User user) {
    boolean b=false;
    String sql="select * from users where username=? and password=?";
    String []parameters={user.getUsername(),user.getPassword()};
    ResultSet rs=SQLHelper.executeQuery(sql, parameters);
    try {
        if(rs.next()) {
            b=true;
        }
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
}

```

```

        return b;
    }
}

```

10、编写 struts.xml

```

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE struts PUBLIC "-//Apache Software Foundation//DTD Struts Configuration 2.1//EN"
"http://struts.apache.org/dtds/struts-2.1.dtd">
<struts>
    <package name="struts2" namespace="/" extends="struts-default">
        <action name="login" class="com.bbc.action.LoginAction">
            <result name="success"/>showbook.jsp</result>
            <result name="input"/>login.jsp</result>
        </action>
        <action name="getPageAction" class="com.bbc.action.GetPageAction">
            <result name="success"/>showbook.jsp</result>
        </action>
        <action name="bookAction" class="com.bbc.action.BookAction">
            <result name="delete"/>deleteok.jsp</result>
            <result name="modify"/>modifyUI.jsp</result>
            <result name="save"/>showbook.jsp</result>
        </action>
    </package>
</struts>

```

11、编写 LoginAction

LoginAction 建议继承于 ActionSupport, 最好使用这种方式, 这样可以使用 ActionSupport 父类中的方法。

```

package com.bbc.action;
import java.util.ArrayList;
import com.bbc.dao.BookDAO;
import com.bbc.dao.UserDAO;
import com.bbc.model.Book;
import com.bbc.model.User;
import com.opensymphony.xwork2.ActionContext;
import com.opensymphony.xwork2.ActionSupport;
public class LoginAction extends ActionSupport {
    private ArrayList<Book> arrayList;
    private User user;
    public ArrayList<Book> getArrayList() {
        return arrayList;
    }
    public void setArrayList(ArrayList<Book> arrayList) {
        this.arrayList = arrayList;
    }
    public User getUser() {

```

```

        return user;
    }

    public void setUser(User user) {
        this.user = user;
    }

    @Override
    public String execute() throws Exception {
        UserDAO userDAO=new UserDAO();
        if(userDAO.checkUser(user)){
            BookDAO bookDAO=new BookDAO();
            ActionContext ac=ActionContext.getContext();
            int pageCount=bookDAO.getPageCount(4);
            //分页 , 第一次显示第一页, 一页显示4行
            ac.put("pageCount", pageCount);
            arrayList=bookDAO.getBookBySelect(1, 4);
            return SUCCESS;
        }else{
            return "input";
        }
    }

    @Override
    public void validate() {
        if(null==user.getUsername()||"".equals(user.getUsername())){
            this.addActionError("用户名不能为空!");
        }
        if(null==user.getPassword()||"".equals(user.getPassword())){
            this.addActionError("密码不能为空!");
        }
    }
}

```

12、编写 BookDAO 类

如果验证通过,就要跳转到 showbook.jsp 页面。在跳到该页面之前,应该把所有的图书信息收集起来,放到一个集合中,以便在 showbook 页面使用循环显示出来。然后在考虑分页技术。所以,首先编写 BookDAO 类。

```

package com.bbc.dao;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import com.bbc.model.Book;
import com.bbc.util.SQLHelper;
public class BookDAO {
    //取出所有的书
    public ArrayList<Book> getAllBook() {
        ArrayList<Book> al=null;
        String sql="select * from book";
    }
}

```



```

String []parameters=null;
al=SQLHelper.executeQueryBook(sql, parameters);
return al;
}
//根据用户点击页码, 返回某一页book的内容
//pageNow:当前页
//pageSize:程序员设定, 表示每页显示多少行
public ArrayList<Book> getBookBySelect(int pageNow, int pageSize) {
    ArrayList<Book> al=new ArrayList<Book>();
    String sql="";
    if(pageNow==1) {
        sql="select top "+pageSize+" * from book";
    } else {
        sql="select top "+pageSize+" * from book where id not in(select top
"+(pageNow-1)*pageSize+" id from book)";
    }
    String []parameters=null;
    ResultSet rs=SQLHelper.executeQuery(sql, parameters);
    try{
        while(rs.next()) {
            Book book=new Book();
            book.setId(rs.getInt(1));
            book.setBookname(rs.getString(2));
            book.setAuthor(rs.getString(3));
            book.setPublishHouse(rs.getString(4));
            book.setPrice(rs.getFloat(5));
            book.setNums(rs.getInt(6));
            al.add(book);
        }
    } catch(SQLException e) {
        e.printStackTrace();
    } finally{
        SQLHelper.close(rs, SQLHelper.getPs(), SQLHelper.getCt());
    }
    return al;
}
//计算分页一共要分多少页 。  分页数= 总记录数/每页的个数   每页个数: pageSize
public int getPageCount(int pageSize) {
    int pageCount=0;
    String sql="select count(*) from book";
    String []parameters=null;
    ResultSet rs=SQLHelper.executeQuery(sql, parameters);
    try {
        rs.next();
        int num=(int)rs.getInt(1);

```

```

        if(num%pageSize==0) {
            pageCount=num/pageSize;
        }else{
            pageCount=num/pageSize+1;
        }
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    return pageCount;
}

//删除一本书
public boolean deleteBook(String id) {
    boolean b=true;
    String sql="delete from book where id=?";
    String []parameters={id};
    try {
        SQLHelper.executeUpdate(sql, parameters);
    } catch (Exception e) {
        b=false;
        e.printStackTrace();
    }
    return b;
}

//取出一本书
public Book getBookByID(String id) {
    Book book=new Book();
    String sql="select * from book where id=?";
    String []parameters={id};
    ResultSet rs=SQLHelper.executeQuery(sql, parameters);
    try {
        if(rs.next()) {
            book.setId(Integer.parseInt(id));
            book.setBookname(rs.getString(2));
            book.setAuthor(rs.getString(3));
            book.setPublishHouse(rs.getString(4));
            book.setPrice(rs.getFloat(5));
            book.setNums(rs.getInt(6));
        }
    } catch (Exception e) {
        e.printStackTrace();
    } finally{
        SQLHelper.close(rs, SQLHelper.getPs(), SQLHelper.getCt());
    }
    return book;
}

```

```

    }
    //更新一本书
    public boolean updateBook(Book book) {
        boolean b=true;
        String sql="update book set bookname=?, author=?, publishHouse=?, price=?, nums=? where
id=?";
        String []parameters={book.getBookname(), book.getAuthor(), book.getPublishHouse(),
String.valueOf(book.getPrice()), String.valueOf(book.getNums()), String.valueOf(book.getId())};
        try {
            SQLHelper.executeUpdate(sql, parameters);
        } catch (Exception e) {
            b=false;
            e.printStackTrace();
        }
        return b;
    }
}

```

13、编写 showbook.jsp 页面

图书管理

[\[Debug\]](#)

编号	书名	作者	出版社	价格	库存	删除	修改
4	jquery技术内幕：深入解析jquery架构设计与实现原理	高云	机械工业出版社	99.0	200	删除	修改
5	轻量级java ee企业应用实战（第3版）	李刚	电子工业出版社	99.0	100	删除	修改
6	Struts+Spring+Hibernate框架及应用开发	王建国	清华大学出版社	69.0	2000	删除	修改
7	Java编程思想(第4版)	Bruce Eckel	机械工业出版社	108.0	3000	删除	修改

[1](#) [2](#)

```

<%@ page language="java" import="java.util.*, com.bbc.model.*" pageEncoding="GB18030"%>
<%
String path = request.getContextPath();
String basePath =
request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";
%>
<%@taglib prefix="s" uri="/struts-tags" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<base href="<%=basePath%>">
<title>图书管理</title>
<meta http-equiv="pragma" content="no-cache">
<meta http-equiv="cache-control" content="no-cache">
<meta http-equiv="expires" content="0">
<meta http-equiv="keywords" content="keyword1, keyword2, keyword3">
<meta http-equiv="description" content="This is my page">

```

```

<!--
<link rel="stylesheet" type="text/css" href="styles.css">
-->
</head>
<body>
<h1>图书管理</h1><br>
<s:debug></s:debug>
<table border=1 bordercolor=green cellpadding=0 id="listdata">
<tr><th>编号</th><th>书名</th><th>作者</th><th>出版社</th><th>价格</th><th>库存
</th><th>删除</th><th>修改</th></tr>
<s:iterator value="arrayList">
<tr>
<td><s:property value="id"/></td>
<td><s:property value="bookname"/></td>
<td><s:property value="author"/></td>
<td><s:property value="publishHouse"/></td>
<td><s:property value="price"/></td>
<td><s:property value="nums"/></td>
<td><a href="/book/bookAction!delete.action?id=${id}">删除</a></td>
<td><a href="/book/bookAction!modify.action?id=${id}">修改</a></td>
</tr>
</s:iterator>
</table>
<% int x= (Integer)request.getAttribute("pageCount"); %>
<% for(int i=1;i<=x;i++)
{
    out.print("<a href=/book/getPageAction.action?pageNow="+i+">"+i+"</a> ");
}
%>
</body>
</html>

```

14、分页 GetPageAction

由于进行了分页，所以进一步编写了分页 GetPageAction

```

package com.bbc.action;
import java.util.ArrayList;
import com.bbc.dao.BookDAO;
import com.bbc.model.Book;
import com.opensymphony.xwork2.ActionContext;
import com.opensymphony.xwork2.ActionSupport;
public class GetPageAction extends ActionSupport {
    private String pageNow;
    private ArrayList<Book> arrayList;
    public ArrayList<Book> getArrayList() {
        return arrayList;
    }
}

```

```

    }
    public void setArrayList(ArrayList<Book> arrayList) {
        this.arrayList = arrayList;
    }
    public String getPageNow() {
        return pageNow;
    }
    public void setPageNow(String pageNow) {
        this.pageNow = pageNow;
    }
    @Override
    public String execute() throws Exception {
        BookDAO bookDAO=new BookDAO();
        int pageCount=bookDAO.getPageCount(4);
        ActionContext.getContext().put("pageCount", pageCount);
        //分页 , 不是第一次分页, 显示第pageNow页, 一页显示4行
        arrayList=bookDAO.getBookBySelect(Integer.parseInt(pageNow), 4);
        return SUCCESS;
    }
}

```

15、编写 bookAction

```

package com.bbc.action;
import java.util.ArrayList;
import com.bbc.dao.BookDAO;
import com.bbc.model.Book;
import com.opensymphony.xwork2.ActionContext;
import com.opensymphony.xwork2.ActionSupport;
public class BookAction extends ActionSupport {
    private String id;
    private Book book;
    private ArrayList<Book> arrayList;
    public ArrayList<Book> getArrayList() {
        return arrayList;
    }
    public void setArrayList(ArrayList<Book> arrayList) {
        this.arrayList = arrayList;
    }
    public Book getBook() {
        return book;
    }
    public void setBook(Book book) {
        this.book = book;
    }
    public String getId() {

```

```
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }

    //增加
    public String add() {
        return SUCCESS;
    }

    //删除
    public String delete() {
        BookDAO bookDAO=new BookDAO();
        bookDAO.deleteBook(id);
        return "delete";
    }

    //修改
    public String modify() {
        BookDAO bookDAO=new BookDAO();
        book=bookDAO.getBookByID(id);
        return "modify";
    }

    //保存
    public String save() {
        //取得当前页面修改后的信息，重新封装成新的book，然后调用updateBook进行更新保存
        BookDAO bookDAO=new BookDAO();
        bookDAO.updateBook(book); //保存成功
        //取出第一页的内容
        int pageCount=bookDAO.getPageCount(4);
        ActionContext.getContext().put("pageCount", pageCount);
        //分页，不是第一次分页，显示第pageNow页，一页显示4行
        arrayList=bookDAO.getBookBySelect(1, 4);
        return "save";
    }
}
```

16、编写 modifyUI.jsp 页面

修改图书信息

[\[Debug\]](#)

图书名称:	jquery技术内幕：深入解析jquery架构设计与实现原理
作者:	高云
出版社:	机械工业出版社
价格:	99.0
库存:	200
<input type="button" value="提交"/>	

```
<%@ page language="java" import="java.util.*" pageEncoding="utf-8"%>
<%@page language="java" import="com.bbc.model.*" %>
<%@taglib prefix="s" uri="/struts-tags"%>
<%
String path = request.getContextPath();
String basePath =
request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";
%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<base href="<%=basePath%>">
<title>My JSP 'deleteUI.jsp' starting page</title>
<meta http-equiv="pragma" content="no-cache">
<meta http-equiv="cache-control" content="no-cache">
<meta http-equiv="expires" content="0">
<meta http-equiv="keywords" content="keyword1,keyword2,keyword3">
<meta http-equiv="description" content="This is my page">
<!--
<link rel="stylesheet" type="text/css" href="styles.css">
-->
</head>
<body>
<h1>修改图书信息</h1> <br>
<s:debug></s:debug>
<form action="bookAction!save.action?id=${id}" method="post" theme="simple">
<s:hidden name="book.id" ></s:hidden>
<table border=1 cellspacing=0 bordercolor=green>
<tr><td>图书名称: </td><td><s:textfield name="book.bookname" theme="simple"
size="50"></s:textfield></td></tr>
<tr><td>作者: </td><td><s:textfield name="book.author" theme="simple"
size="50"></s:textfield></td></tr>
<tr><td>出版社: </td><td><s:textfield name="book.publishHouse" theme="simple"
```

```
size="50"></s:textfield></td></tr>
    <tr><td>价格: </td><td><s:textfield name="book.price" theme="simple"
size="50"></s:textfield></td></tr>
    <tr><td>库存: </td><td><s:textfield name="book.nums" theme="simple"
size="50"></s:textfield></td></tr>
    <tr><td colspan=2 align="center"><input type="submit" value="提交"/></td></tr>
</table>
</form>
</body>
</html>
```

17、增加图书界面以及相应的 action，查找界面以及相应的 action 留给各位同学自行练习!!!