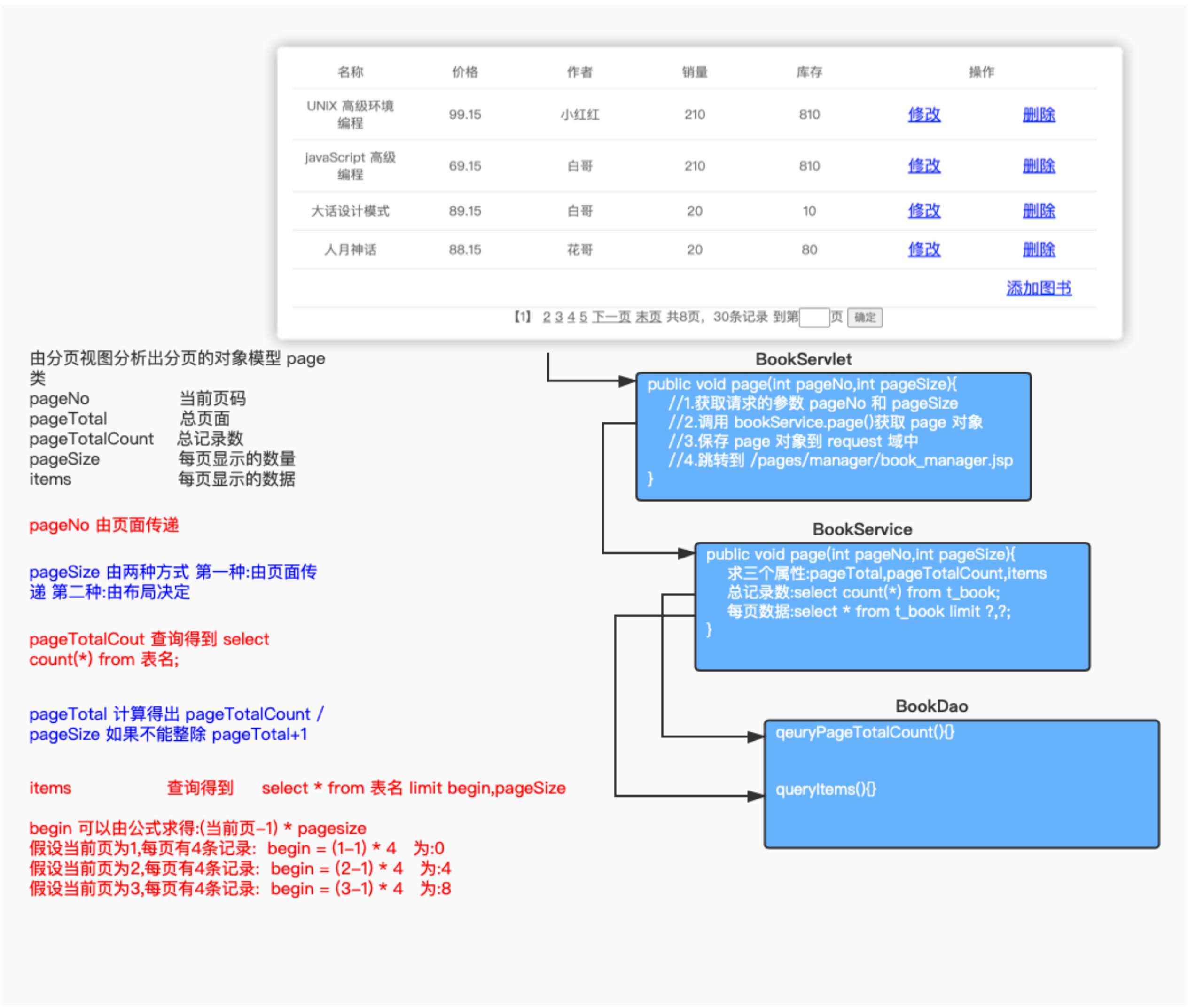
分页

1. 图书分页

1. 分页模块的分析



名称	价格	作者	销量	库存	操作	
UNIX 高级环境编程	99.15	小红红	210	810	修改	删除
javaScript 高级编程	69.15	白哥	210	810	修改	删除
大话设计模式	89.15	白哥	20	10	修改	删除
人月神话	88.15	花哥	20	80	修改	删除
						添加图书

【1】 2 3 4 5 下一页 末页 共8页，30条记录 到第 页 确定

由分页视图分析出分页的对象模型 page 类

pageNo	当前页码
pageTotal	总页面
pageTotalCount	总记录数
pageSize	每页显示的数量
items	每页显示的数据

pageNo 由页面传递

pageSize 由两种方式 第一种:由页面传递 第二种:由布局决定

pageTotalCout 查询得到 select count(*) from 表名;

pageTotal 计算得出 pageTotalCount / pageSize 如果不能整除 pageTotal+1

items 查询得到 select * from 表名 limit begin,pageSize

begin 可以由公式求得:(当前页-1) * pagesize
假设当前页为1,每页有4条记录: begin = (1-1) * 4 为:0
假设当前页为2,每页有4条记录: begin = (2-1) * 4 为:4
假设当前页为3,每页有4条记录: begin = (3-1) * 4 为:8

BookServlet

```
public void page(int pageNo,int pageSize){
    //1.获取请求的参数 pageNo 和 pageSize
    //2.调用 bookService.page()获取 page 对象
    //3.保存 page 对象到 request 域中
    //4.跳转到 /pages/manager/book_manager.jsp
}
```

BookService

```
public void page(int pageNo,int pageSize){
    求三个属性:pageTotal,pageTotalCount,items
    总记录数:select count(*) from t_book;
    每页数据:select * from t_book limit ?,?;
}
```

BookDao

```
qeuryPageTotalCount(){}

queryItems(){}
```

2. 分页模型 Page 的抽取（当前页数，总页数，总记录数， 当前页数据，每页记录数）

Page.java

```java
package com.admiral.pojo;

import java.util.List;

/**
```

```
 * @author 白世鑫
 * @title: Page
 * @projectName JavaWeb
 * @description:
 * @date 2020/9/3  1:20 上午
 */
public class Page<T> {

    public static final Integer PAGE_SIZE = 4;

    //当前页
    private Integer pageNo;
    //总记录数
    private Integer pageTotalCount;
    //总页数
    private Integer pageTotal;
    //当前页显示的数量
    private Integer pageSize = PAGE_SIZE;
    //每页显示的数据
    private List<T> items;

    public Integer getPageNo() {
        return pageNo;
    }

    public void setPageNo(Integer pageNo) {
        this.pageNo = pageNo;
    }

    public Integer getPageTotalCount() {
        return pageTotalCount;
    }

    public void setPageTotalCount(Integer pageTotalCount) {
        this.pageTotalCount = pageTotalCount;
    }

    public Integer getPageTotal() {
        return pageTotal;
    }

    public void setPageTotal(Integer pageTotal) {
        this.pageTotal = pageTotal;
    }

    public Integer getPageSize() {
        return pageSize;
    }
    public void setPageSize(Integer pageSize) {
        this.pageSize = pageSize;
    }

    public List<T> getItems() {
        return items;
    }

    public void setItems(List<T> items) {
        this.items = items;
    }
}
```

3. 分页的初步实现

3.1 在BaseDao中添加查询一行一列的方法

```
    /**
     * 执行返回 一行一列的 sql 语句
     * @param sql 要执行的sql语句
     * @param args 执行sql语句的参数值
     * @return
     */
    public Object queryForSingleValue(String sql,Object... args){
        Connection conn = JdbcUtils.getConnection();

        try {
            return queryRunner.query(conn,sql,new ScalarHandler(),args);
        } catch (SQLException throwables) {
            throwables.printStackTrace();
        } finally {
            JdbcUtils.close(conn);
        }
        return null;
    }
```

BaseDao.java 完整代码如下:

```
package com.admiral.dao;

import com.admiral.utils.JdbcUtils;
```

```java
import org.apache.commons.dbutils.QueryRunner;
import org.apache.commons.dbutils.handlers.BeanHandler;
import org.apache.commons.dbutils.handlers.BeanListHandler;
import org.apache.commons.dbutils.handlers.ScalarHandler;

import java.sql.Connection;
import java.sql.SQLException;
import java.util.List;

/**
 * @author 白世鑫
 * @title: BaseDao
 * @projectName JavaWeb
 * @description:
 * @date 2020/8/27  2:41 上午
 */
public abstract class BaseDao {

    //使用DbUtils 操作数据库
    private QueryRunner queryRunner = new QueryRunner();

    /**
     * 用来执行 inset/update/delete 语句
     * @param sql   要执行的 sql 语句
     * @param args  执行sql语句对应的参数值
     * @return      受影响的行数
     */
    public int update(String sql,Object... args){
        Connection conn = JdbcUtils.getConnection();
        try {
            return queryRunner.update(conn, sql, args);
        } catch (SQLException throwables) {
            throwables.printStackTrace();
        } finally {
            JdbcUtils.close(conn);
        }
        return -1;
    }

    /**
     * 查询返回一个 JavaBean 的方法
     * @param type  返回的对象类型
     * @param sql   要执行的sql语句
     * @param args  sql语句对应的参数值
     * @param <T>   返回类型的泛型
     * @return
     */
    public <T>T queryForOne(Class<T> type, String sql,Object...args){
        Connection conn = JdbcUtils.getConnection();
        try {
            return queryRunner.query(conn,sql,new BeanHandler<T>(type),args);
        } catch (SQLException throwables) {
            throwables.printStackTrace();
        } finally {
            JdbcUtils.close(conn);
        }
        return null;
    }

    /**
     * 执行返回多个 JavaBean 的 sql 语句的方法
     * @param sql
     * @param type
     * @param args
     * @param <T>
     * @return
     */
    public <T>List<T> queryForList(String sql,Class<T> type,Object... args){
        Connection conn = JdbcUtils.getConnection();

        try {
            return queryRunner.query(conn,sql,new BeanListHandler<T>
(type),args);
        } catch (SQLException throwables) {
            throwables.printStackTrace();
        } finally {
            JdbcUtils.close(conn);
        }
        return null;
    }

    /**
     * 执行返回 一行一列的 sql 语句
     * @param sql 要执行的sql语句
     * @param args 执行sql语句的参数值
     * @return
     */
    public Object queryForSingleValue(String sql,Object... args){
        Connection conn = JdbcUtils.getConnection();

        try {
            return queryRunner.query(conn,sql,new ScalarHandler(),args);
        } catch (SQLException throwables) {
            throwables.printStackTrace();
        } finally {
            JdbcUtils.close(conn);
        }
        return null;
    }
}
```

3.2 在BookDao中添加方法

- public Integer queryForPageTotalCount() 方法用户查询总记录数
- public List queryForPageItems() 方法用户查询当前页需要显示的数据

BookDao.java

```java
package com.admiral.dao;


import com.admiral.pojo.Book;

import java.util.List;

/**
 * @author 白世鑫
 * @description
 * @date 2020/9/2
 */
public interface BookDao {

    public int addBook(Book book);

    public int updateBook(Book book);

    public int deleteBookById(Integer id);

    public Book queryBookById(Integer id);

    public List<Book> queryBooks();

    public Integer queryForPageTotalCount();

    public List<Book> queryForPageItems(int begin,int pageSize);
}
```

BookDaoImpl.java

```java
package com.admiral.dao.impl;

import com.admiral.dao.BaseDao;
import com.admiral.dao.BookDao;
import com.admiral.pojo.Book;

import java.util.List;

/**
 * @author 白世鑫
 * @title: BookDaoImpl
 * @projectName Tomcat
 * @description:
 * @date 2020/9/2  1:28 上午
 */
public class BookDaoImpl extends BaseDao implements BookDao {
    @Override
    public int addBook(Book book) {
        String sql = "insert into
t_book(name,price,author,sales,stock,img_path) values(?,?,?,?,?,?)";
        return
update(sql,book.getName(),book.getPrice(),book.getAuthor(),book.getSales(),boo
k.getStock(),book.getImg_path());
    }

    @Override
    public int updateBook(Book book) {
        String sql = "update t_book set
name=?,price=?,author=?,sales=?,stock=?,img_path=? where id=?";
        return
update(sql,book.getName(),book.getPrice(),book.getAuthor(),book.getSales(),boo
k.getStock(),book.getImg_path(),book.getId());
    }

    @Override
    public int deleteBookById(Integer id) {
        String sql = "delete from t_book where id=?";
        return update(sql,id);
    }

    @Override
    public Book queryBookById(Integer id) {
        String sql = "select id,name,price,author,sales,stock,img_path from
t_book where id=?";
        return queryForOne(Book.class,sql,id);
    }

    @Override
    public List<Book> queryBooks() {
        String sql = "select id,name,price,author,sales,stock,img_path from 
t_book";
        return queryForList(sql,Book.class);
    }

    @Override
    public Integer queryForPageTotalCount() {
        String sql = "select count(*) from t_book";
        Number number = (Number) queryForSingleValue(sql);
        return number.intValue();
    }

    @Override
    public List<Book> queryForPageItems(int begin, int pageSize) {
        String sql = "select id,name,price,author,sales,stock,img_path from 
t_book limit ?,?";
        return queryForList(sql,Book.class,begin,pageSize);
    }
}
```

3.3 修改 BookService

- public Page page(int pageNo,int pageSize); 方法负责封装 page 对象

BookService.java

```
package com.admiral.service;

import com.admiral.pojo.Book;
import com.admiral.pojo.Page;

import java.util.List;

/**
 * @author 白世鑫
 * @description
 * @date 2020/9/2
 */
public interface BookService {

    public int addBook(Book book);

    public int updateBook(Book book);

    public int deleteBookById(Integer id);
    public Book queryBookById(Integer id);

    public List<Book> queryBooks();

    public Page<Book> page(int pageNo,int pageSize);

}
```

BookServiceImpl.java

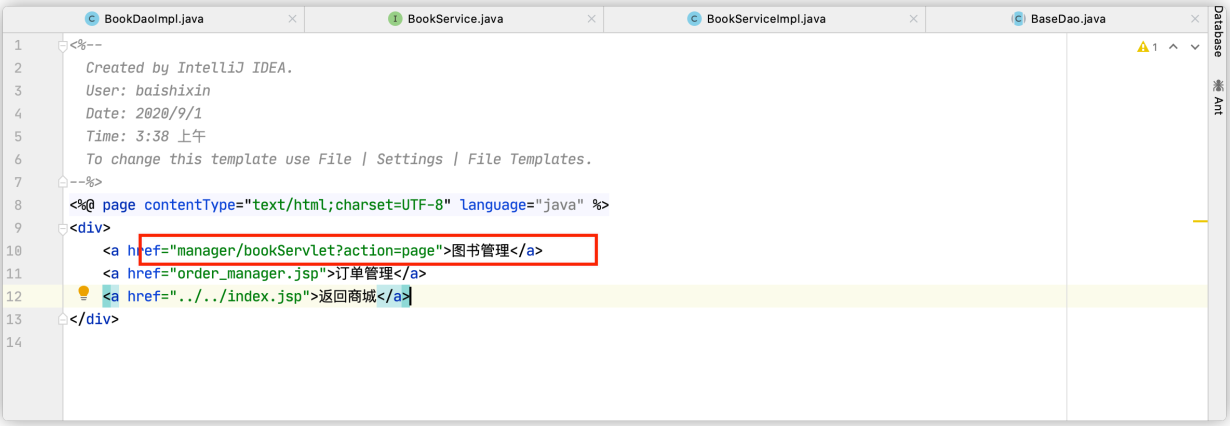
```
package com.admiral.service.impl;


import com.admiral.dao.BookDao;
import com.admiral.dao.impl.BookDaoImpl;
import com.admiral.pojo.Book;
import com.admiral.pojo.Page;
import com.admiral.service.BookService;

import java.util.List;

/**
 * @author 白世鑫
 * @title: BookServiceImpl
 * @projectName Tomcat
 * @description:
 * @date 2020/9/2  1:46 上午
 */
public class BookServiceImpl implements BookService {

    private BookDao bookDao = new BookDaoImpl();

    @Override
    public int addBook(Book book) {
        return bookDao.addBook(book);
    }

    @Override
    public int updateBook(Book book) {
        return bookDao.updateBook(book);
    }

    @Override
    public int deleteBookById(Integer id) {
        return bookDao.deleteBookById(id);
    }
```

```java
    @Override
    public Book queryBookById(Integer id) {
        return bookDao.queryBookById(id);
    }

    @Override
    public List<Book> queryBooks() {
        return bookDao.queryBooks();
    }

    @Override
    public Page<Book> page(int pageNo, int pageSize) {

        Page<Book> page = new Page<>();

        //设置当前页
        page.setPageNo(pageNo);
        //设置每页显示的数量
        page.setPageSize(pageSize);

        //调用 dao 求总记录数
        Integer pageTotalCount = bookDao.queryForPageTotalCount();
        //设置总记录数
        page.setPageTotalCount(pageTotalCount);

        //调用 dao 求每页显示的数据
        int begin = (pageNo - 1) * pageSize;
        List<Book> items = bookDao.queryForPageItems(begin, pageSize);
        //设置每页显示的数据
        page.setItems(items);

        //求总页数
        Integer pageTotal = pageTotalCount / pageSize;
        if (pageTotalCount % pageSize > 0) {
            pageTotal += 1;
        }
        //设置总页数
        page.setPageTotal(pageTotal);

        return page;
    }
}
```

BookServlet.java

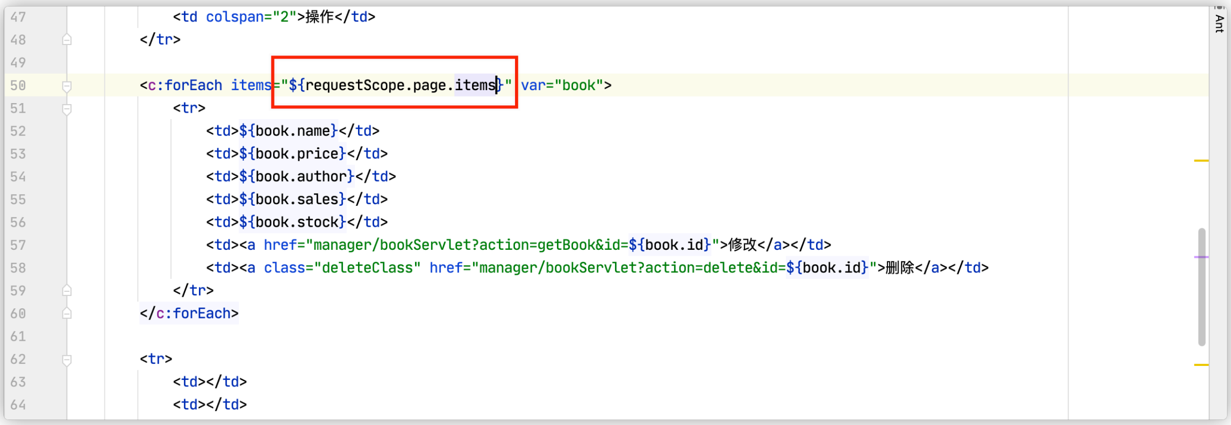
```
package com.admiral.web;


import com.admiral.pojo.Book;
import com.admiral.pojo.Page;
import com.admiral.service.BookService;
import com.admiral.service.impl.BookServiceImpl;
import com.admiral.utils.WebUtils;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.util.List;

/**
 * @author 白世鑫
 * @title: BookServlet
 * @projectName Tomcat
 * @description:
 * @date 2020/9/2  2:05 上午
 */
public class BookServlet extends BaseServlet {

    private BookService bookService = new BookServiceImpl();

    protected void add(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        //1.获取请求参数,封装成 Book 对象
        Book book = WebUtils.copyParamToBean(request, new Book());
        //2.调用 service 保存book对象
        bookService.addBook(book);
        //3.跳到图书列表页面
//          request.getRequestDispatcher("/manager/bookServlet?
action=list").forward(request, response);
        response.sendRedirect(request.getContextPath() +
"/manager/bookServlet?action=list");
    }

    protected void update(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        //1.获取请求参数 封装为 JavaBean
        Book book = WebUtils.copyParamToBean(request, new Book());
        //2.调用 service 更新图书
        bookService.updateBook(book);
        //3.重定向到 图书列表
        response.sendRedirect(request.getContextPath()+"/manager/bookServlet?
action=list");
    }

    protected void getBook(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        //1.获取请求参数 id
        String id = request.getParameter("id");
        //2.根据id调用service查询图书信息
        Book book = bookService.queryBookById(WebUtils.parseInt(id, 0));
        //3.将查询到的图书信息保存到 request 域中
        request.setAttribute("book", book);
        //4.跳转到 book_edit.jsp

 request.getRequestDispatcher("/pages/manager/book_edit.jsp").forward(request,
response);
    }

    protected void delete(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        //1.获取请求的参数  id
        String strId = request.getParameter("id");
        //2.调用service删除图书
        bookService.deleteBookById(WebUtils.parseInt(strId, 0));
        //3.重定向到 /book/manager/bookServlet?action=list
        response.sendRedirect(request.getContextPath() +
"/manager/bookServlet?action=list");
    }

    protected void list(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        //1.查询数据
        List<Book> books = bookService.queryBooks();
        //2.将数据保存到 request 域中
        request.setAttribute("books", books);
        //3.请求转发到 /pages/manager/book_manager.jsp 页面

 request.getRequestDispatcher("/pages/manager/book_manager.jsp").forward(reque
st, response);
    }

    protected void page(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        //1.获取请求的参数  pageNo  pageSize
        int pageNo = WebUtils.parseInt(request.getParameter("pageNo"), 1);
        int pageSize = WebUtils.parseInt(request.getParameter("pageSize"),
Page.PAGE_SIZE);
        //2.调用service获取 page 对象
        Page<Book> page = bookService.page(pageNo, pageSize);
        //3.将 page 对象保存到 request 域中
        request.setAttribute("page",page);
        //4.请求转发到 /pages/manager/book_manager.jsp 页面

 request.getRequestDispatcher("/pages/manager/book_manager.jsp").forward(reque
st, response);
    }
}
```

3.4 修改 manager_menu.jsp



```
<%--
  Created by IntelliJ IDEA.
  User: baishixin
  Date: 2020/9/1
  Time: 3:38 上午
  To change this template use File | Settings | File Templates.
--%>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<div>
    <a href="manager/bookServlet?action=page">图书管理</a>
    <a href="order_manager.jsp">订单管理</a>
    <a href="../../index.jsp">返回商城</a>
</div>
```

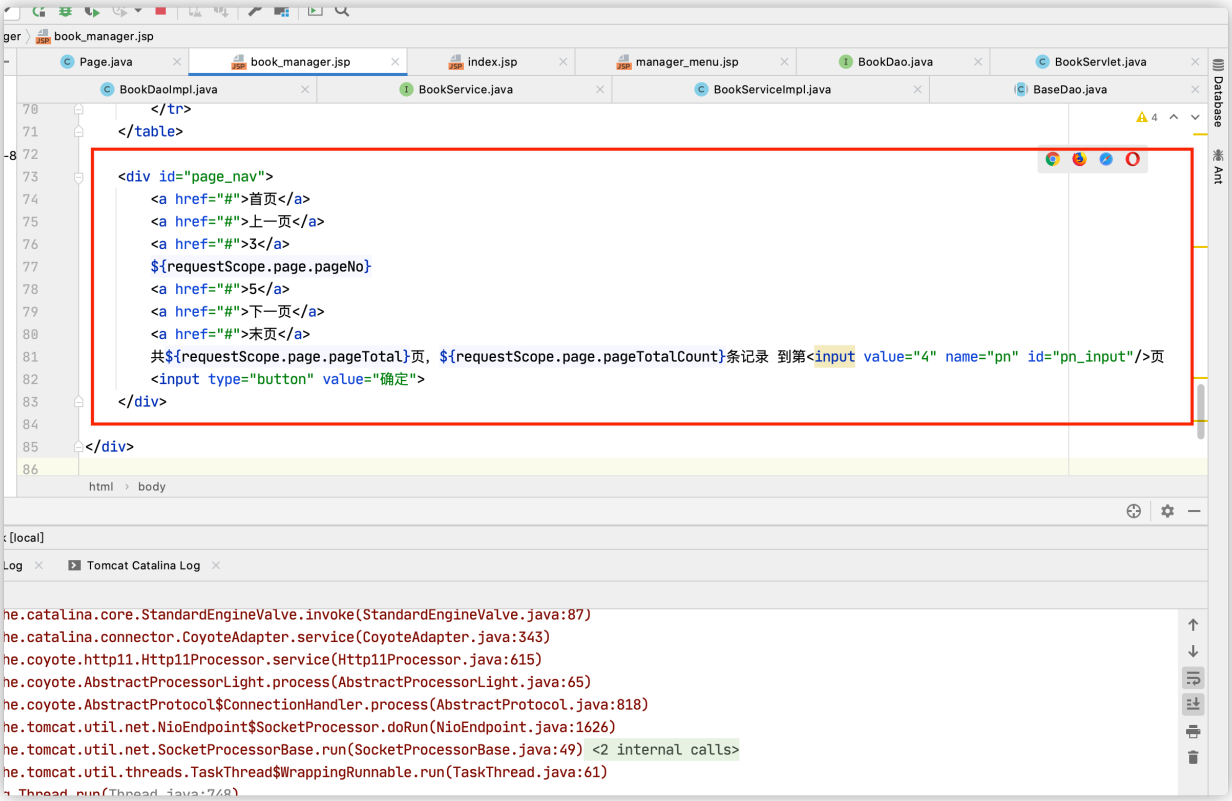
3.5 修改 book_manager.jsp



```
            <td colspan="2">操作</td>
        </tr>

        <c:forEach items="${requestScope.page.items}" var="book">
            <tr>
                <td>${book.name}</td>
                <td>${book.price}</td>
                <td>${book.author}</td>
                <td>${book.sales}</td>
                <td>${book.stock}</td>
                <td><a href="manager/bookServlet?action=getBook&id=${book.id}">修改</a></td>
                <td><a class="deleteClass" href="manager/bookServlet?action=delete&id=${book.id}">删除</a></td>
            </tr>
        </c:forEach>

        <tr>
            <td></td>
            <td></td>
```

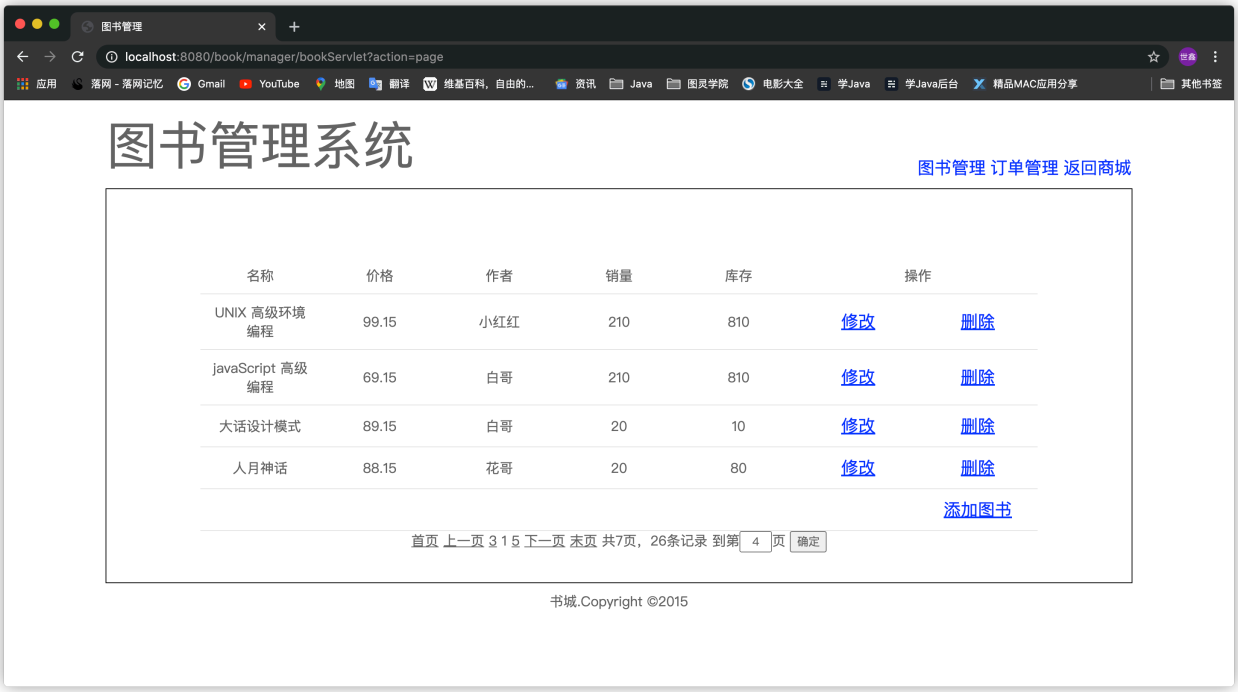
3.6 在 book_manager.jsp 中添加分页条

```
    <div id="page_nav">
        <a href="#">首页</a>
        <a href="#">上一页</a>
        <a href="#">3</a>
        ${requestScope.page.pageNo}
        <a href="#">5</a>
        <a href="#">下一页</a>
        <a href="#">末页</a>
        共${requestScope.page.pageTotal}页，${requestScope.page.pageTotalCount}
条记录 到第<input value="4" name="pn" id="pn_input"/>页
        <input type="button" value="确定">
    </div>
```



```
        </tr>
    </table>

    <div id="page_nav">
        <a href="#">首页</a>
        <a href="#">上一页</a>
        <a href="#">3</a>
        ${requestScope.page.pageNo}
        <a href="#">5</a>
        <a href="#">下一页</a>
        <a href="#">末页</a>
        共${requestScope.page.pageTotal}页，${requestScope.page.pageTotalCount}条记录 到第<input value="4" name="pn" id="pn_input"/>页
        <input type="button" value="确定">
    </div>

</div>
```

```
he.catalina.core.StandardEngineValve.invoke(StandardEngineValve.java:87)
he.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:343)
he.coyote.http11.Http11Processor.service(Http11Processor.java:615)
he.coyote.AbstractProcessorLight.process(AbstractProcessorLight.java:65)
he.coyote.AbstractProtocol$ConnectionHandler.process(AbstractProtocol.java:818)
he.tomcat.util.net.NioEndpoint$SocketProcessor.doRun(NioEndpoint.java:1626)
he.tomcat.util.net.SocketProcessorBase.run(SocketProcessorBase.java:49) <2 internal calls>
he.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:61)
a.Thread.run(Thread.java:748)
```



图书管理系统

图书管理 订单管理 返回商城

名称	价格	作者	销量	库存	操作	
UNIX 高级环境编程	99.15	小红红	210	810	修改	删除
javaScript 高级编程	69.15	白哥	210	810	修改	删除
大话设计模式	89.15	白哥	20	10	修改	删除
人月神话	88.15	花哥	20	80	修改	删除
						添加图书

首页 上一页 3 1 5 下一页 末页 共7页，26条记录 到第 4 页 确定

书城.Copyright ©2015

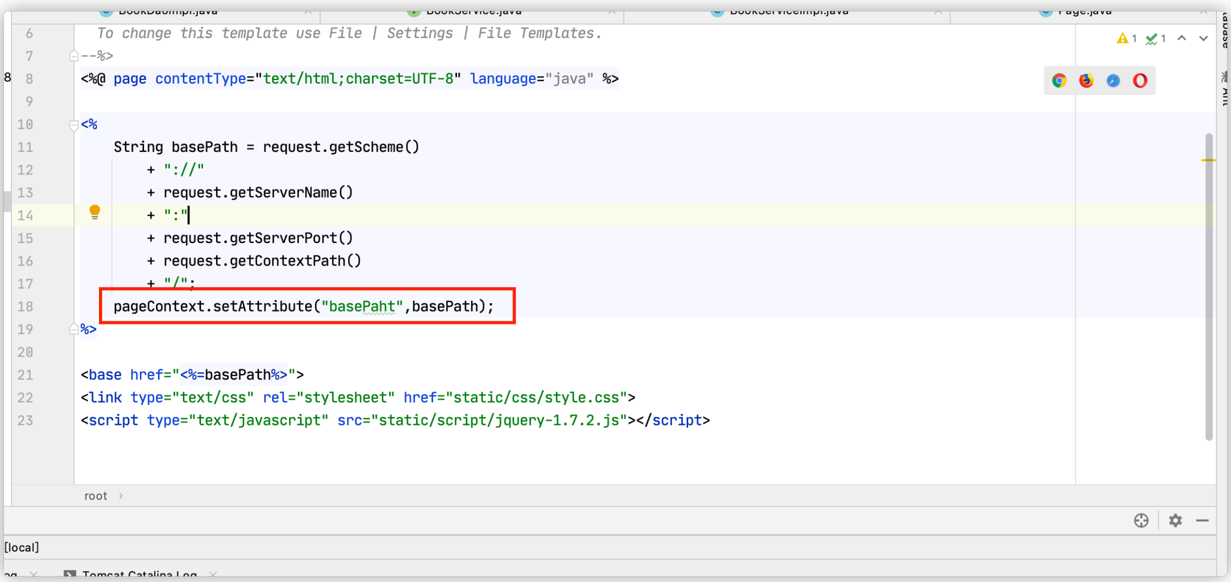
4. 首页、上一页、下一页、末页实现

修改 book_manager.jsp 中的分页条

```
    <div id="page_nav">
        <c:if test="${requestScope.page.pageNo>1}">
            <a href="manager/bookServlet?action=page&pageNo=1">首页</a>
            <a href="manager/bookServlet?
action=page&pageNo=${requestScope.page.pageNo-1}">上一页</a>
        </c:if>
        <a href="#">3</a>
        ${requestScope.page.pageNo}
        <a href="#">5</a>
        <c:if test="${requestScope.page.pageNo<requestScope.page.pageTotal}">
            <a href="manager/bookServlet?
action=page&pageNo=${requestScope.page.pageNo+1}">下一页</a>
            <a href="manager/bookServlet?
action=page&pageNo=${requestScope.page.pageTotal}">末页</a>
        </c:if>
        共${requestScope.page.pageTotal}页，${requestScope.page.pageTotalCount}
条记录 到第<input value="4" name="pn" id="pn_input"/>页
        <input type="button" value="确定">
    </div>
```

5. 分页模块中跳转到指定页数功能实现

header.jsp中的修改



```
    To change this template use File | Settings | File Templates.
--%>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>

<%
    String basePath = request.getScheme()
        + "://"
        + request.getServerName()
        + ":"
        + request.getServerPort()
        + request.getContextPath()
        + "/";
    pageContext.setAttribute("basePaht",basePath);
%>

<base href="<%=basePath%>">
<link type="text/css" rel="stylesheet" href="static/css/style.css">
<script type="text/javascript" src="static/script/jquery-1.7.2.js"></script>
```

添加按钮单击事件

```
<script type="text/javascript">
            $(function () {
                // 跳到指定的页码
                $("#searchPageBtn").click(function () {
                    var pageNo = $("#pn_input").val();
                    <%--var pageTotal = ${requestScope.page.pageTotal};--%>
                    <%--alert(pageTotal);--%>
                    // javaScript 语言中提供了一个 location 地址栏对象
                    // 它有一个属性叫 href.它可以获取浏览器地址栏中的地址
                    // href 属性可读, 可写
                    location.href = "${pageScope.basePath}manager/bookServlet?
action=page&pageNo=" + pageNo;
                });
            });
</script>
```

Page 类的修改

```
    public void setPageNo(Integer pageNo) {
        if (pageNo < 1) {
            pageNo = 1;
        }
        if (pageNo > pageTotal) {
            pageNo = pageTotal;
        }
        this.pageNo = pageNo;
    }
```

BookService中的修改:

```java
    @Override
    public Page<Book> page(int pageNo, int pageSize) {

        Page<Book> page = new Page<>();


        //设置每页显示的数量
        page.setPageSize(pageSize);

        //调用 dao 求总记录数
        Integer pageTotalCount = bookDao.queryForPageTotalCount();
        //设置总记录数
        page.setPageTotalCount(pageTotalCount);


        //求总页数
        Integer pageTotal = pageTotalCount / pageSize;
        if (pageTotalCount % pageSize > 0) {
            pageTotal += 1;
        }
        //设置总页数
        page.setPageTotal(pageTotal);

        //设置当前页
        page.setPageNo(pageNo);

        System.out.println("service中的 pageNo : " + page.getPageNo());

        //调用 dao 求每页显示的数据
        int begin = (page.getPageNo() - 1) * pageSize;
        List<Book> items = bookDao.queryForPageItems(begin, pageSize);
        //设置每页显示的数据
        page.setItems(items);


        return page;
    }
```

6. 分页模块中，页码 1,2,【3】,4,5 的显示，要显示 5 个页 码，并且页码可以点击跳转。

需求：显示 5 个连续的页码，而且当前页码在中间。除了当前页码之外，每个页码都可以点击跳到指定页。

情况 1：如果总页码小于等于 5 的情况，页码的范围是：1-总页码

1 页 1

2 页 1，2

3 页 1，2，3

4 页 1，2，3，4

5 页 1，2，3，4，5

情况 2：总页码大于 5 的情况。假设一共 10 页

小情况 1：当前页码为前面 3 个：1，2，3 的情况，页码范围是：1-5.

[1] 2，3，4，5

1 [2] 3，4，5

1 2 [3] 4，5

小情况 2：当前页码为最后 3 个，8，9，10，页码范围是：总页码减 4 - 总页码

6，7【8】9，10

6，7，8【9】10

6，7，8，9【10】

小情况 3：4，5，6，7，页码范围是：当前页码减 2 - 当前页码加 2

2，3，4，5，6

3，4，5，6，7

4，5，6，7，8

5，6，7，8，9

修改 book_manager.jsp

```
        <c:choose>
            <%--情况 1：如果总页码小于等于 5 的情况，页码的范围是：1-总页码--%>
            <c:when test="${requestScope.page.pageTotal<=5}">
                <c:set var="begin" value="1" />
                <c:set var="end" value="${requestScope.page.pageTotal}" />

            </c:when>
            <%--情况 2：总页码大于 5 的情况。假设一共 10 页--%>
            <c:when test="${requestScope.page.pageTotal>5}">
                <c:choose>
                    <%--小情况 1：当前页码为前面 3 个：1，2，3 的情况，页码范围是：1-
5.--%>
                    <c:when test="${requestScope.page.pageNo <= 3}">
                        <c:set var="begin" value="1" />
                        <c:set var="end" value="5" />

                    </c:when>
                    <%--小情况 2：当前页码为最后 3 个，8，9，10，页码范围是：总页码减 4
- 总页码--%>
                    <c:when test="${requestScope.page.pageNo >
requestScope.page.pageTotal-3}">
                        <c:set var="begin"
value="${requestScope.page.pageTotal-4}" />
                        <c:set var="end"
value="${requestScope.page.pageTotal}" />

                    </c:when>
                    <%--小情况 3：4，5，6，7，页码范围是：当前页码减 2 - 当前页码加 2-
-%>
                    <c:otherwise>
                        <c:set var="begin" value="${requestScope.page.pageNo-
2}" />
                        <c:set var="end" value="${requestScope.page.pageNo+2}"
/>

                    </c:otherwise>
                </c:choose>
            </c:when>
        </c:choose>

        <c:forEach begin="${begin}" end="${end}" var="i">
            <c:if test="${requestScope.page.pageNo == i}">
                ${i}
            </c:if>
            <c:if test="${requestScope.page.pageNo != i}">
                <a href="manager/bookServlet?action=page&pageNo=${i}">${i}</a>
            </c:if>
        </c:forEach>
```

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>图书管理</title>
    <%--  静态包含 base css样式 jQuery  --%>
    <%@ include file="/pages/common/header.jsp" %>

    <script type="text/javascript">
        $(function () {
            // 给删除的 a 标签绑定单击事件，用于删除的确认提示操作
            $("a.deleteClass").click(function () {
                // 在事件的 function 函数中，有一个 this 对象。这个 this 对象，是当前
正在响应事件的 dom 对象。
                /**
                 * confirm 是确认提示框函数
                 * 参数是它的提示内容
                 * 它有两个按钮，一个确认，一个是取消。
                 * 返回 true 表示点击了，确认，返回 false 表示点击取消。
                 */
                return confirm("你确定要删除【" +
$(this).parent().parent().find("td:first").text() + "】?");
                // return false// 阻止元素的默认行为===不提交请求
            });
        });
    </script>


</head>
<body>

<div id="header">
    <img class="logo_img" alt="" src="../../static/img/logo.gif">
    <span class="wel_word">图书管理系统</span>
    <%-- 静态包含管理操作菜单 --%>
    <%@ include file="/pages/common/manager_menu.jsp" %>
</div>

<div id="main">
    <table>
```

```
        <tr>
            <td>名称</td>
            <td>价格</td>
            <td>作者</td>
            <td>销量</td>
            <td>库存</td>
            <td colspan="2">操作</td>
        </tr>

        <c:forEach items="${requestScope.page.items}" var="book">
            <tr>
                <td>${book.name}</td>
                <td>${book.price}</td>
                <td>${book.author}</td>
                <td>${book.sales}</td>
                <td>${book.stock}</td>
                <td><a href="manager/bookServlet?
action=getBook&id=${book.id}">修改</a></td>
                <td><a class="deleteClass" href="manager/bookServlet?
action=delete&id=${book.id}">删除</a></td>
            </tr>
        </c:forEach>

        <tr>
            <td></td>
            <td></td>
            <td></td>
            <td></td>
            <td></td>
            <td></td>
            <td><a href="pages/manager/book_edit.jsp">添加图书</a></td>
        </tr>
    </table>

    <div id="page_nav">
        <c:if test="${requestScope.page.pageNo>1}">
            <a href="manager/bookServlet?action=page&pageNo=1">首页</a>
            <a href="manager/bookServlet?
action=page&pageNo=${requestScope.page.pageNo-1}">上一页</a>
        </c:if>


        <c:choose>
            <%--情况 1: 如果总页码小于等于 5 的情况，页码的范围是: 1-总页码--%>
            <c:when test="${requestScope.page.pageTotal<=5}">
                <c:set var="begin" value="1" />
                <c:set var="end" value="${requestScope.page.pageTotal}" />

            </c:when>
            <%--情况 2：总页码大于 5 的情况。假设一共 10 页--%>
            <c:when test="${requestScope.page.pageTotal>5}">
                <c:choose>
                    <%--小情况 1：当前页码为前面 3 个：1，2，3 的情况，页码范围是：1-
5.--%>
                    <c:when test="${requestScope.page.pageNo <= 3}">
                        <c:set var="begin" value="1" />
                        <c:set var="end" value="5" />

                    </c:when>
                    <%--小情况 2：当前页码为最后 3 个，8，9，10，页码范围是：总页码减 4
- 总页码--%>
                    <c:when test="${requestScope.page.pageNo >
requestScope.page.pageTotal-3}">
                        <c:set var="begin"
value="${requestScope.page.pageTotal-4}" />
                        <c:set var="end"
value="${requestScope.page.pageTotal}" />

                    </c:when>
                    <%--小情况 3：4，5，6，7，页码范围是：当前页码减 2 - 当前页码加 2-
-%>
                    <c:otherwise>
                        <c:set var="begin" value="${requestScope.page.pageNo-
2}" />
                        <c:set var="end" value="${requestScope.page.pageNo+2}"
/>

                    </c:otherwise>
                </c:choose>
            </c:when>
        </c:choose>

        <c:forEach begin="${begin}" end="${end}" var="i">
            <c:if test="${requestScope.page.pageNo == i}">
                ${i}
            </c:if>
            <c:if test="${requestScope.page.pageNo != i}">
                <a href="manager/bookServlet?action=page&pageNo=${i}">${i}</a>
            </c:if>
        </c:forEach>


        <c:if test="${requestScope.page.pageNo<requestScope.page.pageTotal}">
            <a href="manager/bookServlet?
action=page&pageNo=${requestScope.page.pageNo+1}">下一页</a>
            <a href="manager/bookServlet?
action=page&pageNo=${requestScope.page.pageTotal}">末页</a>
        </c:if>
        共${requestScope.page.pageTotal}页，${requestScope.page.pageTotalCount}
条记录
        到第<input value="${param.pageNo}" name="pn" id="pn_input"/>页
        <input id="searchPageBtn" type="button" value="确定">

        <script type="text/javascript">
            $(function () {
                // 跳到指定的页码
                $("#searchPageBtn").click(function () {
                    var pageNo = $("#pn_input").val();
                    <%--var pageTotal = ${requestScope.page.pageTotal};--%>
<%--alert(pageTotal);--%>
                    // javaScript 语言中提供了一个 location 地址栏对象
                    // 它有一个属性叫 href.它可以获取浏览器地址栏中的地址
                    // href 属性可读，可写
                    location.href = "${pageScope.basePath}manager/bookServlet?
action=page&pageNo=" + pageNo;
                });
            });
        </script>
    </div>

</div>

<%-- 静态包含页脚 --%>
<%@ include file="/pages/common/footer.jsp" %>
</body>
</html>
```

7. 修改分页后，增加，删除，修改图书信息的回显页面

添加

修改 book_manager.jsp 中添加图书请求

```
<td><a href="pages/manager/book_edit.jsp?
pageNo=${requestScope.page.pageTotal}">添加图书</a></td>
```

在 book_edit.jsp 页面中添加隐藏域

```
<input type="hidden" name="pageNo" value="${param.pageNo}">
```

修改 BookServlet 中 add方法的跳转路径

```
    protected void add(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        Integer pageNo = WebUtils.parseInt(request.getParameter("pageNo"),0);
        pageNo+=1;

        //1.获取请求参数,封装成 Book 对象
        Book book = WebUtils.copyParamToBean(request, new Book());
        //2.调用 service 保存book对象
        bookService.addBook(book);
        //3.跳到图书列表页面
//          request.getRequestDispatcher("/manager/bookServlet?
action=list").forward(request, response);
        response.sendRedirect(request.getContextPath() +
"/manager/bookServlet?action=page&pageNo="+pageNo);
    }
```

将 BookServlet 中所有 action=list 的请求替换为 action=page

删除

修改删除按钮的请求地址

```
<td><a class="deleteClass" href="manager/bookServlet?
action=delete&id=${book.id}&pageNo=${requestScope.page.pageNo}">删除</a></td>
```

修改 BookServlet 中删除的跳转

```
    protected void delete(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        //1.获取请求的参数  id
        String strId = request.getParameter("id");
        //2.调用service删除图书
        bookService.deleteBookById(WebUtils.parseInt(strId, 0));
        //3.重定向到 /book/manager/bookServlet?action=list
        response.sendRedirect(request.getContextPath() +
"/manager/bookServlet?action=page&pageNo="+request.getParameter("pageNo"));
    }
```

修改

修改 update 按钮的请求

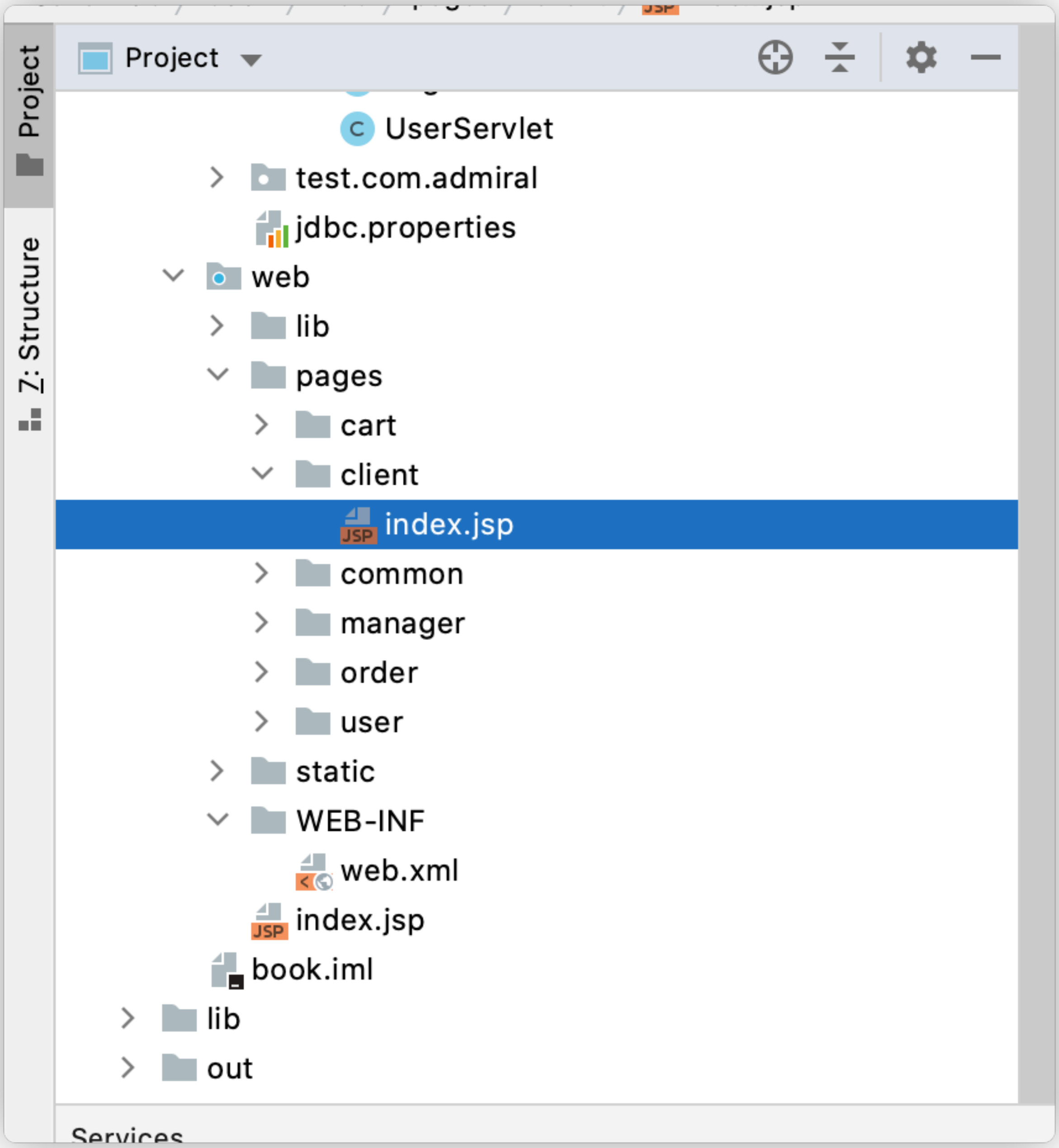
```
<td><a href="manager/bookServlet?
action=getBook&id=${book.id}&pageNo=${requestScope.page.pageNo}">修改</a></td>
```

修改 BookServlet 中修改后的跳转

```
    protected void update(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        //1.获取请求参数 封装为 JavaBean
        Book book = WebUtils.copyParamToBean(request, new Book());
        //2.调用 service 更新图书
        bookService.updateBook(book);
        //3.重定向到 图书列表
        response.sendRedirect(request.getContextPath()+"/manager/bookServlet?
action=page&pageNo="+request.getParameter("pageNo"));
    }
```

2. 首页 index.jsp 的跳转

在 web/pages 目录下新建 client 目录,然后将 index.jsp 复制到 web/pages/client 下



新建 ClientBookServlet

```
package com.admiral.web;

import com.admiral.pojo.Book;
import com.admiral.pojo.Page;
import com.admiral.service.BookService;
import com.admiral.service.impl.BookServiceImpl;
import com.admiral.utils.WebUtils;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
```

```java
/**
 * @author 白世鑫
 * @title: ClientBookServlet
 * @projectName JavaWeb
 * @description:
 * @date 2020/9/3  4:25 上午
 */
public class ClientBookServlet extends BaseServlet{

    private BookService bookService = new BookServiceImpl();

    protected void page(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        //1.获取请求的参数  pageNo  pageSize
        int pageNo = WebUtils.parseInt(request.getParameter("pageNo"), 1);
        int pageSize = WebUtils.parseInt(request.getParameter("pageSize"),
Page.PAGE_SIZE);
        //2.调用service获取 page 对象
        Page<Book> page = bookService.page(pageNo, pageSize);
        //3.将 page 对象保存到 request 域中
        request.setAttribute("page",page);
        //4.请求转发到 /pages/client/index.jsp 页面

 request.getRequestDispatcher("/pages/client/index.jsp").forward(request,
response);
    }

}
```

配置 web.xml

```xml
    <servlet>
        <servlet-name>ClientBookServlet</servlet-name>
        <servlet-class>com.admiral.web.ClientBookServlet</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>ClientBookServlet</servlet-name>
        <url-pattern>/client/bookServlet</url-pattern>
    </servlet-mapping>
```

在 index.jsp 中做请求转发

```jsp
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<jsp:forward page="/client/bookServlet?action=page"></jsp:forward>
```

在 pages/client/index.jsp中 遍历数据

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>书城首页</title>
    <%--  静态包含 base css样式 jQuery  --%>
  <%@ include file="/pages/common/header.jsp"%>
</head>
<body>

<div id="header">
    <img class="logo_img" alt="" src="static/img/logo.gif">
    <span class="wel_word">网上书城</span>
    <div>
        <a href="pages/user/login.jsp">登录</a> |
        <a href="pages/user/regist.jsp">注册</a> &nbsp;&nbsp;
        <a href="pages/cart/cart.jsp">购物车</a>
        <a href="pages/manager/manager.jsp">后台管理</a>
    </div>
</div>
<div id="main">


    <div id="book">
        <div class="book_cond">
            <form action="" method="get">
                价格: <input id="min" type="text" name="min" value=""> 元 -
                <input id="max" type="text" name="max" value=""> 元
                <input type="submit" value="查询"/>
            </form>
        </div>
        <div style="text-align: center">
            <span>您的购物车中有3件商品</span>
            <div>
                您刚刚将<span style="color: red">时间简史</span>加入到了购物车中
            </div>
        </div>

        <%--开始--%>
        <c:forEach items="${requestScope.page.items}" var="book">
        <div class="b_list">
            <div class="img_div">
                <img class="book_img" alt="" src="${book.img_path}"/>
            </div>
            <div class="book_info">
                <div class="book_name">
                    <span class="sp1">书名:</span>
                    <span class="sp2">${book.name}</span>
                </div>
                <div class="book_author">
                    <span class="sp1">作者:</span>
                    <span class="sp2">${book.author}</span>
                </div>
                <div class="book_price">
                    <span class="sp1">价格:</span>
                    <span class="sp2">￥${book.price}</span>
                </div>
                <div class="book_sales">
                    <span class="sp1">销量:</span>
                    <span class="sp2">${book.sales}</span>
                </div>
                <div class="book_amount">
                    <span class="sp1">库存:</span>
                    <span class="sp2">${book.stock}</span>
                </div>
                <div class="book_add">
                    <button>加入购物车</button>
                </div>
            </div>
        </div>
        </c:forEach>
        <%--结束--%>
    </div>


</div>
<%-- 静态包含页脚 --%>
<%@ include file="/pages/common/footer.jsp"%>
</body>
</html>
```

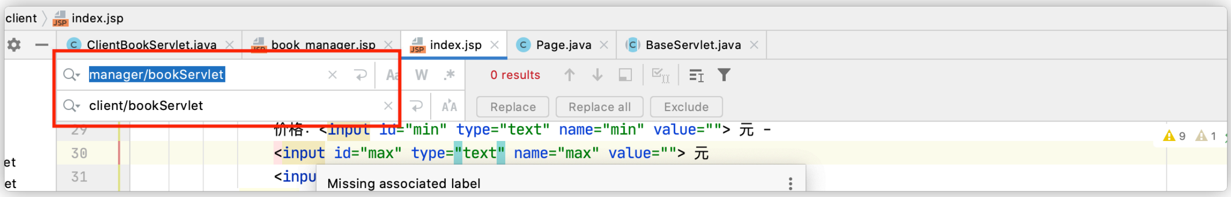
将后台的分页拷贝到 前台的 index.jsp中

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>书城首页</title>
    <%--  静态包含 base css样式 jQuery  --%>
  <%@ include file="/pages/common/header.jsp"%>
</head>
<body>

<div id="header">
    <img class="logo_img" alt="" src="static/img/logo.gif">
    <span class="wel_word">网上书城</span>
    <div>
        <a href="pages/user/login.jsp">登录</a> |
        <a href="pages/user/regist.jsp">注册</a> &nbsp;&nbsp;
        <a href="pages/cart/cart.jsp">购物车</a>
        <a href="pages/manager/manager.jsp">后台管理</a>
    </div>
</div>
<div id="main">


    <div id="book">
        <div class="book_cond">
            <form action="" method="get">
                价格：<input id="min" type="text" name="min" value=""> 元 -
                <input id="max" type="text" name="max" value=""> 元
                <input type="submit" value="查询"/>
            </form>
        </div>
        <div style="text-align: center">
            <span>您的购物车中有3件商品</span>
            <div>
                您刚刚将<span style="color: red">时间简史</span>加入到了购物车中
            </div>
        </div>

        <%--开始--%>
        <c:forEach items="${requestScope.page.items}" var="book">
        <div class="b_list">
            <div class="img_div">
                <img class="book_img" alt="" src="${book.img_path}"/>
            </div>
            <div class="book_info">
                <div class="book_name">
                    <span class="sp1">书名:</span>
                    <span class="sp2">${book.name}</span>
                </div>
                <div class="book_author">
                    <span class="sp1">作者:</span>
                    <span class="sp2">${book.author}</span>
                </div>
                <div class="book_price">
```

```
                    <span class="sp1">价格:</span>
                    <span class="sp2">￥${book.price}</span>
                </div>
                <div class="book_sales">
                    <span class="sp1">销量:</span>
                    <span class="sp2">${book.sales}</span>
                </div>
                <div class="book_amount">
                    <span class="sp1">库存:</span>
                    <span class="sp2">${book.stock}</span>
                </div>
                <div class="book_add">
                    <button>加入购物车</button>
                </div>
            </div>
        </div>
        </c:forEach>
        <%--结束--%>
    </div>

    <div id="page_nav">
        <c:if test="${requestScope.page.pageNo>1}">
            <a href="client/bookServlet?action=page&pageNo=1">首页</a>
            <a href="client/bookServlet?
action=page&pageNo=${requestScope.page.pageNo-1}">上一页</a>
        </c:if>


        <c:choose>
            <%--情况 1：如果总页码小于等于 5 的情况，页码的范围是：1-总页码--%>
            <c:when test="${requestScope.page.pageTotal<=5}">
                <c:set var="begin" value="1" />
                <c:set var="end" value="${requestScope.page.pageTotal}" />

            </c:when>
            <%--情况 2：总页码大于 5 的情况。假设一共 10 页--%>
            <c:when test="${requestScope.page.pageTotal>5}">
                <c:choose>
                    <%--小情况 1：当前页码为前面 3 个：1，2，3 的情况，页码范围是：1-
5.--%>
                    <c:when test="${requestScope.page.pageNo <= 3}">
                        <c:set var="begin" value="1" />
                        <c:set var="end" value="5" />

                    </c:when>
                    <%--小情况 2：当前页码为最后 3 个，8，9，10，页码范围是：总页码减 4
- 总页码--%>
                    <c:when test="${requestScope.page.pageNo >
requestScope.page.pageTotal-3}">
```

```
                        <c:set var="begin"
value="${requestScope.page.pageTotal-4}" />
                        <c:set var="end"
value="${requestScope.page.pageTotal}" />

                    </c:when>
                    <%--小情况 3: 4, 5, 6, 7, 页码范围是: 当前页码减 2 - 当前页码加 2-
-%>
                    <c:otherwise>
                        <c:set var="begin" value="${requestScope.page.pageNo-
2}" />
                        <c:set var="end" value="${requestScope.page.pageNo+2}"
/>

                    </c:otherwise>
                </c:choose>
            </c:when>
        </c:choose>

        <c:forEach begin="${begin}" end="${end}" var="i">
            <c:if test="${requestScope.page.pageNo == i}">
                【${i}】
            </c:if>
            <c:if test="${requestScope.page.pageNo != i}">
                <a href="client/bookServlet?action=page&pageNo=${i}">${i}</a>
            </c:if>
        </c:forEach>


        <c:if test="${requestScope.page.pageNo<requestScope.page.pageTotal}">
            <a href="client/bookServlet?
action=page&pageNo=${requestScope.page.pageNo+1}">下一页</a>
            <a href="client/bookServlet?
action=page&pageNo=${requestScope.page.pageTotal}">末页</a>
        </c:if>
        共${requestScope.page.pageTotal}页, ${requestScope.page.pageTotalCount}
条记录
        到第<input value="${param.pageNo}" name="pn" id="pn_input"/>页
        <input id="searchPageBtn" type="button" value="确定">

        <script type="text/javascript">
            $(function () {
                // 跳到指定的页码
                $("#searchPageBtn").click(function () {
                    var pageNo = $("#pn_input").val();
                    <%--var pageTotal = ${requestScope.page.pageTotal};--%>
<%--alert(pageTotal);--%>
                    // javaScript 语言中提供了一个 location 地址栏对象
                    // 它有一个属性叫 href.它可以获取浏览器地址栏中的地址
                    // href 属性可读，可写
                    location.href = "${pageScope.basePath}client/bookServlet?
action=page&pageNo=" + pageNo;
                });
            });
        </script>
    </div>
</div>
<%-- 静态包含页脚 --%>
<%@ include file="/pages/common/footer.jsp"%>
</body>
</html>
```

修改所有 manager/bookServlet 为 client/bookServlet



```
价格：<input id="min" type="text" name="min" value=""> 元 -
<input id="max" type="text" name="max" value=""> 元
<inpu
```

3. 分页条的抽取

3.1 抽取分页条中请求地址为 url 变量

3.1.1 在 page 对象中添加 url 属性

3.1.2 在 Servlet 程序的 page 分页方法中设置 url 的分页请求地址

3.1.3 修改分页条中请求地址为 url 变量输出,并抽取一个单独的 jsp 页面

4. 首页价格搜索