

# 12--项目第五阶段-图书分页

讲师：王振国

## 今日任务

## 2、图书分页

### 1)分页模块的分析

木虚肉盖饭	16.00	小胖	1000	50				
C++编程思想	45.50	刚哥	14	95				
蛋炒饭	9.90	周星星	12	赌神	66.50	龙伍	125	535
赌神	66.50	龙伍	125	535				

由分页的视图分析出分页的对象模型Page类

```

pageNo      当前页码
pageTotal   总页码
pageTotalCount 总记录数
pageSize    每页显示数量
items       当前页数据
    
```

pageNo 当前页码是由客户端进行传递

pageSize 每页显示数量由两种因素决定。  
一：客户端进行传递  
二：由页面布局决定

pageTotalCount 总记录数可以由sql语句求得。  
sql语句是：select count(\*) from 表名。

pageTotal 总页码可以由总记录数/每页数量得到。  
注：总记录数%每页数量>0,则 总页码+1

items 是当前页数据，也是可以由sql语句求得。  
sql语句是：select \* from 表名 limit begin ,pageSize;

begin 可以由公式求得：(pageNo-1) x pageSize;  
假设，当前是第1页，每页4条记录，则begin = (1-1) x 4 得：0  
假设，当前是第2页，每页4条记录，则begin = (2-1) x 4 得：4  
假设，当前是第3页，每页4条记录，则begin = (3-1) x 4 得：8

需要传递两个参数：  
pageNo  
和  
pageSize

BookServlet程序

```

public void page() { 处理分页
1、获取请求的参数 pageNo 和 pageSize。
2、调用BookService.page(pageNo, pageSize):Page对象
3、保存到Request域中
4、请求转发到/pages/manager/book_manager.jsp页面
}
    
```

BookService程序

```

public Page page(pageNo, pageSize) { 处理分页业务
求三个属性即可：总记录数、总页码、当前页数据
总记录数：select count(*) from 表名
当前页数据：select * from 表名 limit begin ,pageSize;
}
    
```

BookDao程序

```

queryForPageTotalCount() 求总记录数
select count(*) from 表名。
queryForItems() 求当前页数据
select * from 表名 limit begin ,pageSize;
    
```

### 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）

```
* @param <T> 是具体的模块的 javaBean 类
*/
public class Page<T> {
```

```
    public static final Integer PAGE_SIZE = 4;
    // 当前页码
    private Integer pageNo;
    // 总页码
    private Integer pageTotal;
    // 当前页显示数量
    private Integer pageSize = PAGE_SIZE;
    // 总记录数
    private Integer pageTotalCount;
    // 当前页数据
    private List<T> items;
```

### 3) 分页的初步实现

BookDao 代码:

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

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

BookService 代码:

```
@Override
public Page<Book> page(int pageNo, int pageSize) {
    Page<Book> page = new Page<Book>();
    // 设置当前页码
    page.setPageNo(pageNo);

    // 设置每页显示的数量
    page.setPageSize(pageSize);
    // 求总记录数
    Integer pageTotalCount = bookDao.queryForPageTotalCount();
    // 设置总记录数
```

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

// 求当前页数据的开始索引
int begin = (page.getPageNo() - 1) * pageSize;
// 求当前页数据
List<Book> items = bookDao.queryForPageItems(begin, pageSize);
// 设置当前页数据
page.setItems(items);

return page;
}
```

BookServlet 程序的代码:

```
/**
 * 处理分页功能
 * @param req
 * @param resp
 * @throws ServletException
 * @throws IOException
 */
protected void page(HttpServletRequest req, HttpServletResponse resp) throws ServletException,
IOException {
    //1 获取请求的参数 pageNo 和 pageSize
    int pageNo = WebUtils.parseInt(req.getParameter("pageNo"), 1);
    int pageSize = WebUtils.parseInt(req.getParameter("pageSize"), Page.PAGE_SIZE);
    //2 调用 BookService.page(pageNo, pageSize): Page 对象
    Page<Book> page = bookService.page(pageNo, pageSize);
    //3 保存 Page 对象到 Request 域中
    req.setAttribute("page", page);
    //4 请求转发到 pages/manager/book_manager.jsp 页面
    req.getRequestDispatcher("/pages/manager/book_manager.jsp").forward(req, resp);
}
```

manager\_menu.jsp 中【图书管理】请求地址的修改:

```
<%@ 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>
```

book\_manager.jsp 修改:

```
<%@ 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 标签、css 样式、jQuery 文件 -->
<%@ include file="/pages/common/head.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">
    
    <span class="wel_word">图书管理系统</span>

    <!-- 静态包含 manager 管理模块的菜单 -->
```

```
<%@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">
```

```
<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>

<!-- 静态包含页脚内容-->
<%@include file="/pages/common/footer.jsp"%>

</body>
</html>
```

## 4) 首页、上一页、下一页、末页实现

```
<div id="page_nav">
  <!-- 大于首页，才显示-->
  <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>
  【${ requestScope.page.pageNo }】
  <a href="#">5</a>
  <!-- 如果已经 是最后一页，则不显示下一页，末页 -->
  <c:if test="${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

```
<div id="page_nav">
  <!-- 大于首页，才显示-->
  <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>
【${ requestScope.page.pageNo }】
<a href="#">5</a>
<!-- 如果已经 是最后一页，则不显示下一页，末页 -->
<c:if test="${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>
```

Page 对象中的修改:

```
public void setPageNo(Integer pageNo) {
    /* 数据边界的有效检查 */
    if (pageNo < 1) {
        pageNo = 1;
    }
    if (pageNo > pageTotal) {
        pageNo = pageTotal;
    }

    this.pageNo = pageNo;
}
```

BookService 中 page 方法的修改:

@Override

```
public Page<Book> page(int pageNo, int pageSize) {
    Page<Book> page = new Page<Book>();
    // 设置每页显示的数量
    page.setPageSize(pageSize);
    // 求总记录数
    Integer pageTotalCount = bookDao.queryForPageTotalCount();
    // 设置总记录数
    page.setPageTotalCount(pageTotalCount);
    // 求总页码
    Integer pageTotal = pageTotalCount / pageSize;
    if (pageTotalCount % pageSize > 0) {
        pageTotal+=1;
    }
    // 设置总页码
    page.setPageTotal(pageTotal);

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

    // 求当前页数据的开始索引
    int begin = (page.getPageNo() - 1) * pageSize;
    // 求当前页数据
    List<Book> items = bookDao.queryForPageItems(begin, pageSize);
    // 设置当前页数据
    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

```
<!-- 页码输出的开始-->
<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-->
  <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="${i == requestScope.page.pageNo}">
        【${i}】
    </c:if>
    <c:if test="${i != requestScope.page.pageNo}">
        <a href="manager/bookServlet?action=page&pageNo=${i}">${i}</a>
    </c:if>
</c:forEach>
<!-- 页码输出的结束-->

```

## 7) 修改分页后，增加，删除，修改图书信息的回显页面

以修改图书为例：

- 1、在修改的请求地址上追加当前页码参数：

```

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

```

- 2、在 book\_edit.jsp 页面中使用隐藏域记录下 pageNo 参数

```

<div id="main">
    <form action="manager/bookServlet" method="get">
        <input type="hidden" name="pageNo" value="${param.pageNo}">
        <input type="hidden" name="action" value="${ empty param.id ? "add" : "update" }" />
        <input type="hidden" name="id" value="${ requestScope.book.id }" />
        <table>
            <tr>

```

- 3、在服务器重定向的时候，获取当前页码追加上进行跳转

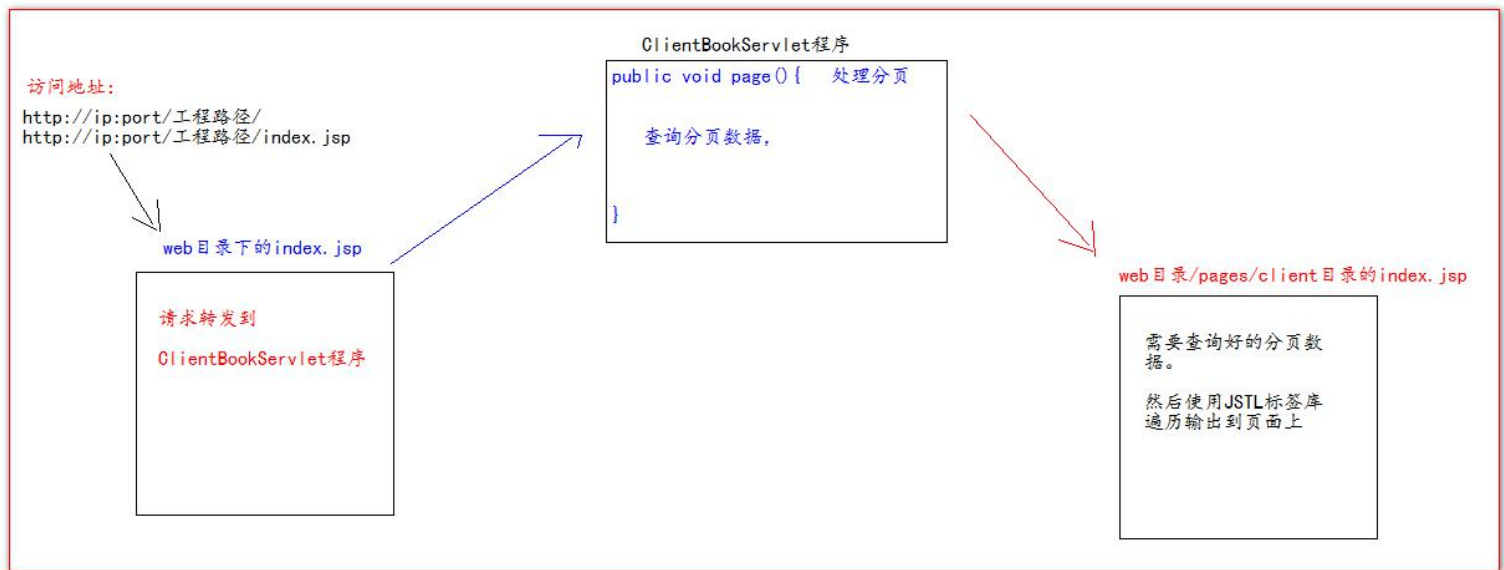
```

protected void update(HttpServletRequest req, HttpServletResponse resp) throws ServletException,
IOException {
    // 1、获取请求的参数==封装成为 Book 对象
    Book book = WebUtils.copyParamToBean(req.getParameterMap(), new Book());
    // 2、调用 BookService.updateBook( book ); 修改图书
    bookService.updateBook(book);
}

```

```
//      3、重定向回图书列表管理页面
//      地址: /工程名/manager/bookServlet?action=list
resp.sendRedirect(req.getContextPath() + "/manager/bookServlet?action=page&pageNo=" +
req.getParameter("pageNo"));
}
```

## 3、首页 index.jsp 的跳转



## 4、分页条的抽取

### 4.1、抽取分页条中请求地址为 url 变量

#### 4.1.1.在 page 对象中添加 url 属性

```
/**
 * Page 是分页的模型对象
 * @param <T> 是具体的模块的 javaBean 类
 */
public class Page<T> {
    public static final Integer PAGE_SIZE = 4;
    // 当前页码
    private Integer pageNo;
    // 总页码
    private Integer pageTotal;
```

```
// 当前页显示数量
private Integer pageSize = PAGE_SIZE;
// 总记录数
private Integer pageTotalCount;
// 当前页数据
private List<T> items;
// 分页条的请求地址
private String url;
```

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

```
protected void page(HttpServletRequest req, HttpServletResponse resp) throws ServletException {
    //1 获取请求的参数 pageNo 和 pageSize
    int pageNo = WebUtils.parseInt(req.getParameter(s: "pageNo"), defaultValue: 1);
    int pageSize = WebUtils.parseInt(req.getParameter(s: "pageSize"), Page.PAGE_SIZE);
    //2 调用BookService.page(pageNo, pageSize): Page对象
    Page<Book> page = bookService.page(pageNo, pageSize);
    page.setUrl("client/bookServlet?action=page");
    //3 保存Page对象到Request域中
    req.setAttribute(s: "page", page);
    //4 请求转发到pages/manager/book_manager.jsp页面
    req.getRequestDispatcher(s: "/pages/client/index.jsp").forward(req, resp);
}
```

#### 4.1.3、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

```
<!-- 分页条的开始-->
<div id="page_nav">
    <!-- 大于首页, 才显示-->
    <c:if test="${requestScope.page.pageNo > 1}">
        <a href="${requestScope.page.url}&pageNo=1">首页</a>
        <a href="${requestScope.page.url}&pageNo=${requestScope.page.pageNo-1}">上一页</a>
    </c:if>
    <!-- 页码输出的开始-->
    <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-->
        <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="${i == requestScope.page.pageNo}">
```

```
        【${i}】
```

```
    </c:if>
```

```
    <c:if test="${i != requestScope.page.pageNo}">
```

```
        <a href="${requestScope.page.url}&pageNo=${i}">${i}</a>
```

```
    </c:if>
```

```
</c:forEach>
```

```
<!-- 页码输出的结束-->
```

```
<!-- 如果已经 是最后一页, 则不显示下一页, 末页 -->
```

```
<c:if test="${requestScope.page.pageNo < requestScope.page.pageTotal}">
```

```
    <a href="${requestScope.page.url}&pageNo=${requestScope.page.pageNo+1}">下一页</a>
```

```
    <a href="${requestScope.page.url}&pageNo=${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 = "${pageScope.basePath}${ requestScope.page.url }&pageNo=" + pageNo;
```

```
});
```

```
});
```

```
</script>
```

```
</div>
```

```
<!-- 分页条的结束-->
```

## 5、首页价格搜索

