1. 项目第六阶段：购物车

1.1 购物车模块分析

1.2 购物车模型编写

1.2.1 购物车模型

购物车商品项:CartItem.java

```
package com.admiral.pojo;

import java.math.BigDecimal;

/**
 * @author 白世鑫
 * @title: CartItem
 * @projectName JavaWeb
 * @description: 购物车商品项
 * @date 2020/9/8  1:43 上午
 */
public class CartItem {
    private Integer id;
    private String name;
    private Integer count;
    private BigDecimal price;
    private BigDecimal totalPrice;

    public CartItem() {
    }

    public CartItem(Integer id, String name, Integer count, BigDecimal price,
BigDecimal totalPrice) {
        this.id = id;
        this.name = name;
        this.count = count;
        this.price = price;
        this.totalPrice = totalPrice;
    }

    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 Integer getCount() {
        return count;
    }

    public void setCount(Integer count) {
        this.count = count;
    }

    public BigDecimal getPrice() {
        return price;
    }

    public void setPrice(BigDecimal price) {
        this.price = price;
    }

    public BigDecimal getTotalPrice() {
        return totalPrice;
    }

    public void setTotalPrice(BigDecimal totalPrice) {
        this.totalPrice = totalPrice;
    }

    @Override
    public String toString() {
        return "CartItem{" +
                "id=" + id +
                ", name='" + name + '\'' +
                ", count=" + count +
                ", price=" + price +
                ", totalPrice=" + totalPrice +
                '}';
    }
}
```

购物车模型对象:Cart.java

```
package com.admiral.pojo;

import java.math.BigDecimal;
import java.util.ArrayList;
import java.util.List;

/**
 * @author 白世鑫
 * @title: Cart
 * @projectName JavaWeb
 * @description: 购物车模型对象
 * @date 2020/9/8  1:45 上午
 */
public class Cart {

    private Integer totalCount;
    private BigDecimal totalPrice;
    private List<CartItem> items = new ArrayList<>();

    public Integer getTotalCount() {
        return totalCount;
    }

    public void setTotalCount(Integer totalCount) {
        this.totalCount = totalCount;
    }

    public BigDecimal getTotalPrice() {
        return totalPrice;
    }

    public void setTotalPrice(BigDecimal totalPrice) {
        this.totalPrice = totalPrice;
    }

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

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

    @Override
    public String toString() {
        return "Cart{" +
                "totalCount=" + totalCount +
```

```
                ", totalPrice=" + totalPrice +
                ", items=" + items +
                '}';
    }
}
```

1.2.2 购物车的测试

在购物车模型中添加增删改查的方法

- addItem()
- deleteItem()
- clear()
- updateItem()

```
package com.admiral.pojo;

import java.math.BigDecimal;
import java.util.LinkedHashMap;
import java.util.Map;

/**
 * @author 白世鑫
 * @title: Cart
 * @projectName JavaWeb
 * @description: 购物车模型对象
 * @date 2020/9/8  1:45 上午
 */
public class Cart {

//    private Integer totalCount;
//    private BigDecimal totalPrice;
//    private List<CartItem> items = new ArrayList<>();
    private Map<Integer,CartItem> items = new LinkedHashMap<Integer, CartItem>
();


    /**
     * 添加商品项
     * @param cartItem
     */
    public void addItems(CartItem cartItem){
        CartItem item = items.get(cartItem.getId());
        if (item == null) {
            //购物车中没有
            items.put(cartItem.getId(),cartItem);
```

```
        }else {
            //购物车中已存在,累加数量和总金额
            //更新数量.累加
            item.setCount(item.getCount()+1);
            //更新总金额
            item.setTotalPrice(item.getPrice().multiply(new
BigDecimal(item.getCount())));
        }
    }

    /**
     * 删除商品项
     * @param id
     */
    public void delteItem(Integer id){
        items.remove(id);
    }

    /**
     * 情况购物车
     */
    public void clear(){
        items.clear();
    }

    /**
     * 更新上数量
     * @param id
     * @param count
     */
    public void updateCount(Integer id,Integer count){
        //先查看购物车中是否有此商品.如果有,更新商品数量,更新总金额.
        CartItem cartItem = items.get(id);
        if (cartItem != null) {
            cartItem.setCount(count);
            cartItem.setTotalPrice(cartItem.getPrice().multiply(new
BigDecimal(cartItem.getCount())));
        }
    }



    public Integer getTotalCount() {
        Integer totalCount = 0;

        for (CartItem cartItem : items.values()) {
            totalCount+=cartItem.getCount();
        }
```

```
        return totalCount;
    }



    public BigDecimal getTotalPrice() {
        BigDecimal totalPrice = new BigDecimal(0);

        for (CartItem cartItem : items.values()) {
            totalPrice = totalPrice.add(cartItem.getTotalPrice());
        }

        return totalPrice;
    }


    public Map<Integer, CartItem> getItems() {
        return items;
    }

    public void setItems(Map<Integer, CartItem> items) {
        this.items = items;
    }

    @Override
    public String toString() {
        return "Cart{" +
                "totalCount=" + getTotalCount() +
                ", totalPrice=" + getTotalPrice() +
                ", items=" + items +
                '}';
    }
}
```

测试类

```
package test.com.admiral.pojo;

import com.admiral.pojo.Cart;
import com.admiral.pojo.CartItem;
import org.junit.Test;

import java.math.BigDecimal;

/**
 * Cart Tester.
 *
 * @author <Authors name>
```

```
 * @version 1.0
 * @since <pre>9月 8, 2020</pre>
 */
public class CartTest {


    /**
     * Method: addItems(CartItem cartItem)
     */
    @Test
    public void testAddItems() throws Exception {
        Cart cart = new Cart();

        cart.addItems(new CartItem(1,"时间简史",1,new BigDecimal(100),new
BigDecimal(100)));
        cart.addItems(new CartItem(1,"时间简史",1,new BigDecimal(100),new
BigDecimal(100)));
        cart.addItems(new CartItem(2,"钢铁是怎样炼成的",1,new BigDecimal(200),new
BigDecimal(200)));

        System.out.println(cart);

    }

    /**
     * Method: delteItem(Integer id)
     */
    @Test
    public void testDelteItem() throws Exception {
        Cart cart = new Cart();

        cart.addItems(new CartItem(1,"时间简史",1,new BigDecimal(100),new
BigDecimal(100)));
        cart.addItems(new CartItem(1,"时间简史",1,new BigDecimal(100),new
BigDecimal(100)));
        cart.addItems(new CartItem(2,"钢铁是怎样炼成的",1,new BigDecimal(200),new
BigDecimal(200)));

        cart.delteItem(1);

        System.out.println(cart);
    }

    /**
     * Method: clear()
     */
    @Test
    public void testClear() throws Exception {
        Cart cart = new Cart();
```

```
        cart.addItems(new CartItem(1,"时间简史",1,new BigDecimal(100),new
BigDecimal(100)));
        cart.addItems(new CartItem(1,"时间简史",1,new BigDecimal(100),new
BigDecimal(100)));
        cart.addItems(new CartItem(2,"钢铁是怎样炼成的",1,new BigDecimal(200),new
BigDecimal(200)));

        cart.delteItem(1);

        cart.clear();

        System.out.println(cart);
    }

    /**
     * Method: updateCount(Integer id, Integer count)
     */
    @Test
    public void testUpdateCount() throws Exception {
        Cart cart = new Cart();

        cart.addItems(new CartItem(1,"时间简史",1,new BigDecimal(100),new
BigDecimal(100)));
        cart.addItems(new CartItem(1,"时间简史",1,new BigDecimal(100),new
BigDecimal(100)));
        cart.addItems(new CartItem(2,"钢铁是怎样炼成的",1,new BigDecimal(200),new
BigDecimal(200)));

        cart.delteItem(1);

        cart.clear();

        cart.addItems(new CartItem(1,"时间简史",1,new BigDecimal(100),new
BigDecimal(100)));
        cart.updateCount(1,10);

        System.out.println(cart);
    }
}
```

1.3 加入购物车功能的实现

CartServlet 程序中的代码

```
package com.admiral.web;
```

```java
import com.admiral.pojo.Book;
import com.admiral.pojo.Cart;
import com.admiral.pojo.CartItem;
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;

/**
 * @author 白世鑫
 * @title: CartServlet
 * @projectName JavaWeb
 * @description:
 * @date 2020/9/8  2:48 上午
 */
public class CartServlet extends BaseServlet {

    private BookService bookService = new BookServiceImpl();

    protected void addItem(HttpServletRequest req, HttpServletResponse resp)
throws ServletException, IOException {
        //获取请求的参数,图书编号
        Integer id = WebUtils.parseInt(req.getParameter("id"), 0);
        //根据编号调动 service 查询图书信息
        Book book = bookService.queryBookById(id);
        //根据图书信息封装cartItem
        CartItem cartItem = new CartItem(book.getId(), book.getName(), 1,
book.getPrice(), book.getPrice());
        //调用 cart.additem 添加到购物车
        Cart cart = (Cart) req.getSession().getAttribute("cart");
        if (cart == null) {
            cart = new Cart();
            req.getSession().setAttribute("cart",cart);
        }
        cart.addItems(cartItem);
        System.out.println(cartItem);
        //重定向到首页
//        resp.sendRedirect(req.getContextPath());
        //重定向到原来的页面
        resp.sendRedirect(req.getHeader("Referer"));

    }
}
```

index.jsp 页面 js 的代码

```
    <script type="text/javascript">
        $(function (){
            $("button.addToCart").click(function () {

                var bookId = $(this).attr("bookId");

                location.href="http://localhost:8080/book/cartServlet?
action=addItem&id="+bookId;
            });
        });
    </script>
```

```
            <span class="sp2">${book.stock}</span>
        </div>
        <div class="book_add">
            <button bookId=${book.id} class="addToCart">加入购物车</button>
        </div>
    </div>
</div>
</c:forEach>
```

1.4 购物车的展示

/pages/cart/cart.jsp

```
<%@ 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>购物车</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">购物车</span>
    <%--  静态包含欢迎信息  --%>
    <%@ include file="/pages/common/success_menu.jsp" %>
</div>

<div id="main">
```

```
    <table>
        <tr>
            <td>商品名称</td>
            <td>数量</td>
            <td>单价</td>
            <td>金额</td>
            <td>操作</td>
        </tr>

        <c:if test="${empty sessionScope.cart.items}">
            <tr>
                <td colspan="5"><a href="index.jsp">亲,购物车空空如也!快去和小伙伴
一起剁手吧!</a></td>
            </tr>
        </c:if>
        <c:if test="${not empty sessionScope.cart.items}">
            <c:forEach items="${sessionScope.cart.items}" var="entry">
                <tr>
                    <td>${entry.value.name}</td>
                    <td>${entry.value.count}</td>
                    <td>${entry.value.price}</td>
                    <td>${enrty.value.totalPrice}</td>
                    <td><a href="#">删除</a></td>
                </tr>
            </c:forEach>
        </c:if>


    </table>
    <c:if test="${not empty sessionScope.cart.items}">
        <div class="cart_info">
            <span class="cart_span">购物车中共有<span
class="b_count">${sessionScope.cart.totalCount}</span>件商品</span>
            <span class="cart_span">总金额<span
class="b_price">${sessionScope.cart.totalPrice}</span>元</span>
            <span class="cart_span"><a href="#">清空购物车</a></span>
            <span class="cart_span"><a href="pages/cart/checkout.jsp">去结账
</a></span>
        </div>
    </c:if>
</div>

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

1.5 删除购物车商品项

CartServlet.java

```
package com.admiral.web;

import com.admiral.pojo.Book;
import com.admiral.pojo.Cart;
import com.admiral.pojo.CartItem;
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;

/**
 * @author 白世鑫
 * @title: CartServlet
 * @projectName JavaWeb
 * @description:
 * @date 2020/9/8  2:48 上午
 */
public class CartServlet extends BaseServlet {

    private BookService bookService = new BookServiceImpl();

    /**
     * 删除购物车商品项
     * @param req
     * @param resp
     * @throws ServletException
     * @throws IOException
     */
    protected void deleteItem(HttpServletRequest req, HttpServletResponse
resp) throws ServletException, IOException {
        int id = WebUtils.parseInt(req.getParameter("id"), 0);

        //先获取购物车对象
        Cart cart = (Cart) req.getSession().getAttribute("cart");

        if (cart != null) {
            cart.delteItem(id);
            //重定向会原来的页面
            resp.sendRedirect(req.getHeader("Referer"));
        }
```

```
    }


    protected void addItem(HttpServletRequest req, HttpServletResponse resp)
throws ServletException, IOException {
        //获取请求的参数,图书编号
        Integer id = WebUtils.parseInt(req.getParameter("id"), 0);
        //根据编号调动 service 查询图书信息
        Book book = bookService.queryBookById(id);
        //根据图书信息封装cartItem
        CartItem cartItem = new CartItem(book.getId(), book.getName(), 1,
book.getPrice(), book.getPrice());
        //调用 cart.additem 添加到购物车
        Cart cart = (Cart) req.getSession().getAttribute("cart");
        if (cart == null) {
            cart = new Cart();
            req.getSession().setAttribute("cart",cart);
        }
        cart.addItems(cartItem);
        System.out.println(cartItem);
        //重定向到首页
//        resp.sendRedirect(req.getContextPath());
        //重定向到原来的页面
        resp.sendRedirect(req.getHeader("Referer"));

    }
}
```

删除的请求地址

```
<td><a class="deleteItem" href="cartServlet?
action=deleteItem&id=${entry.value.id}">删除</a></td>
```

删除的单击事件:

```
    <script type="text/javascript">
        $(function () {
            $("a.deleteItem").click(function () {
                return confirm("你确定要删除【" +
$(this).parent().parent().find("td:first").text() +"】吗?");
            })
        });
    </script>
```

Cart.jsp

```
<%@ 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>购物车</title>
    <%--  静态包含 base css样式 jQuery  --%>
    <%@ include file="/pages/common/header.jsp" %>

    <script type="text/javascript">
        $(function () {
            $("a.deleteItem").click(function () {
                return confirm("你确定要删除【" +
$(this).parent().parent().find("td:first").text() +"】吗?");
            })
        });
    </script>

</head>
<body>


<div id="header">
    <img class="logo_img" alt="" src="static/img/logo.gif">
    <span class="wel_word">购物车</span>
    <%--  静态包含欢迎信息  --%>
    <%@ include file="/pages/common/success_menu.jsp" %>
</div>

<div id="main">

    <table>
        <tr>
            <td>商品名称</td>
            <td>数量</td>
            <td>单价</td>
            <td>金额</td>
            <td>操作</td>
        </tr>

        <c:if test="${empty sessionScope.cart.items}">
            <tr>
                <td colspan="5"><a href="index.jsp">亲,购物车空空如也!快去和小伙伴
一起剁手吧!</a></td>
            </tr>
        </c:if>
        <c:if test="${not empty sessionScope.cart.items}">
            <c:forEach items="${sessionScope.cart.items}" var="entry">
```

```
                <tr>
                    <td>${entry.value.name}</td>
                    <td>${entry.value.count}</td>
                    <td>${entry.value.price}</td>
                    <td>${enrty.value.totalPrice}</td>
                    <td><a class="deleteItem" href="cartServlet?
action=deleteItem&id=${entry.value.id}">删除</a></td>
                </tr>
            </c:forEach>
        </c:if>


    </table>
    <c:if test="${not empty sessionScope.cart.items}">
        <div class="cart_info">
            <span class="cart_span">购物车中共有<span
class="b_count">${sessionScope.cart.totalCount}</span>件商品</span>
            <span class="cart_span">总金额<span
class="b_price">${sessionScope.cart.totalPrice}</span>元</span>
            <span class="cart_span"><a href="#">清空购物车</a></span>
            <span class="cart_span"><a href="pages/cart/checkout.jsp">去结账
</a></span>
        </div>
    </c:if>
</div>

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

1.6 清空购物车

```
package com.admiral.web;

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

import javax.servlet.ServletException;
```

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

/**
 * @author 白世鑫
 * @title: CartServlet
 * @projectName JavaWeb
 * @description:
 * @date 2020/9/8  2:48 上午
 */
public class CartServlet extends BaseServlet {

    private BookService bookService = new BookServiceImpl();


    protected void clear(HttpServletRequest req, HttpServletResponse resp)
throws ServletException, IOException {
        //1.先获取购物车对象
        Cart cart = (Cart) req.getSession().getAttribute("cart");
        if (cart != null) {
            cart.clear();
            //从哪里来回哪里去
            resp.sendRedirect(req.getHeader("Referer"));
        }

    }
    /**
     * 删除购物车商品项
     * @param req
     * @param resp
     * @throws ServletException
     * @throws IOException
     */
    protected void deleteItem(HttpServletRequest req, HttpServletResponse
resp) throws ServletException, IOException {
        int id = WebUtils.parseInt(req.getParameter("id"), 0);

        //先获取购物车对象
        Cart cart = (Cart) req.getSession().getAttribute("cart");

        if (cart != null) {
            cart.delteItem(id);
            //重定向会原来的页面
            resp.sendRedirect(req.getHeader("Referer"));
        }

    }
```

```
    protected void addItem(HttpServletRequest req, HttpServletResponse resp)
throws ServletException, IOException {
        //获取请求的参数,图书编号
        Integer id = WebUtils.parseInt(req.getParameter("id"), 0);
        //根据编号调动 service 查询图书信息
        Book book = bookService.queryBookById(id);
        //根据图书信息封装cartItem
        CartItem cartItem = new CartItem(book.getId(), book.getName(), 1,
book.getPrice(), book.getPrice());
        //调用 cart.additem 添加到购物车
        Cart cart = (Cart) req.getSession().getAttribute("cart");
        if (cart == null) {
            cart = new Cart();
            req.getSession().setAttribute("cart",cart);
        }
        cart.addItems(cartItem);
        System.out.println(cartItem);
        //重定向到首页
//        resp.sendRedirect(req.getContextPath());
        //重定向到原来的页面
        resp.sendRedirect(req.getHeader("Referer"));

    }
}
```

```
<%@ 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>购物车</title>
    <%--  静态包含 base css样式 jQuery  --%>
    <%@ include file="/pages/common/header.jsp" %>

    <script type="text/javascript">
        $(function () {
            $("a.deleteItem").click(function () {
                return confirm("你确定要删除【" +
$(this).parent().parent().find("td:first").text() + "】吗?");
            })

            // 给清空购物车绑定单击事件
            $("#clearCart").click(function () {
                return confirm("你确定要清空购物车吗?");
```

```
            })
        });
    </script>

</head>
<body>


<div id="header">
    <img class="logo_img" alt="" src="static/img/logo.gif">
    <span class="wel_word">购物车</span>
    <%--  静态包含欢迎信息  --%>
    <%@ include file="/pages/common/success_menu.jsp" %>
</div>

<div id="main">

    <table>
        <tr>
            <td>商品名称</td>
            <td>数量</td>
            <td>单价</td>
            <td>金额</td>
            <td>操作</td>
        </tr>

        <c:if test="${empty sessionScope.cart.items}">
            <tr>
                <td colspan="5"><a href="index.jsp">亲,购物车空空如也!快去和小伙伴
一起剁手吧!</a></td>
            </tr>
        </c:if>
        <c:if test="${not empty sessionScope.cart.items}">
            <c:forEach items="${sessionScope.cart.items}" var="entry">
                <tr>
                    <td>${entry.value.name}</td>
                    <td>${entry.value.count}</td>
                    <td>${entry.value.price}</td>
                    <td>${enrty.value.totalPrice}</td>
                    <td><a class="deleteItem" href="cartServlet?
action=deleteItem&id=${entry.value.id}">删除</a></td>
                </tr>
            </c:forEach>
        </c:if>


    </table>
    <c:if test="${not empty sessionScope.cart.items}">
        <div class="cart_info">
```

```
            <span class="cart_span">购物车中共有<span
class="b_count">${sessionScope.cart.totalCount}</span>件商品</span>
            <span class="cart_span">总金额<span
class="b_price">${sessionScope.cart.totalPrice}</span>元</span>
            <span class="cart_span"><a id="clearCart" href="cartServlet?
action=clear">清空购物车</a></span>
            <span class="cart_span"><a href="pages/cart/checkout.jsp">去结账
</a></span>
        </div>
    </c:if>
</div>

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

1.7 修改购物车商品数量

事件

```
            // 给输入框绑定 onchange 内容发生改变事件
            $(".updateCount").change(function () {
                // 获取商品名称
                var name = $(this).parent().parent().find("td:first").text();
                var id = $(this).attr('bookId');
                // 获取商品数量
                var count = this.value;
                if (confirm("你确定要将【" + name + "】商品修改数量为: " + count +
" 吗?")) {
                    //发起请求。给服务器保存修改
                    location.href = "http://localhost:8080/book/cartServlet?
action=updateCount&count=" + count + "&id=" + id;
                } else {
                    // defaultValue 属性是表单项 Dom 对象的属性。它表示默认的 value
属性值。
                    this.value = this.defaultValue;
                }
            });
```

```
    <c:if test="${not empty sessionScope.cart.items}">
        <c:forEach items="${sessionScope.cart.items}" var="entry">
            <tr>
                <td>${entry.value.name}</td>
                <td>
                    <input bookId="${entry.value.id}" type="text" style="width: 80px"
                           class="updateCount" value="${entry.value.count}">
                </td>
                <td>${entry.value.price}</td>
                <td>${enrty.value.totalPrice}</td>
                <td><a class="deleteItem" href="cartServlet?action=deleteItem&id=${entry.value.id}">删除</a></td>
            </tr>
        </c:forEach>
    </c:if>


</table>
```

cart.jsp

```
<%@ 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>购物车</title>
    <%--  静态包含 base css样式 jQuery  --%>
    <%@ include file="/pages/common/header.jsp" %>

    <script type="text/javascript">
        $(function () {
            $("a.deleteItem").click(function () {
                return confirm("你确定要删除【" +
$(this).parent().parent().find("td:first").text() + "】吗?");
            })

            // 给清空购物车绑定单击事件
            $("#clearCart").click(function () {
                return confirm("你确定要清空购物车吗?");
            })

            // 给输入框绑定 onchange 内容发生改变事件
            $(".updateCount").change(function () {
                // 获取商品名称
                var name = $(this).parent().parent().find("td:first").text();
                var id = $(this).attr('bookId');
                // 获取商品数量
                var count = this.value;
                if (confirm("你确定要将【" + name + "】商品修改数量为: " + count +
" 吗?")) {
```

```
                    //发起请求。给服务器保存修改
                    location.href = "http://localhost:8080/book/cartServlet?action=updateCount&count=" + count + "&id=" + id;
                } else {
                    // defaultValue 属性是表单项 Dom 对象的属性。它表示默认的 value 属性值。
                    this.value = this.defaultValue;
                }
            });
        });
    </script>

</head>
<body>


<div id="header">
    <img class="logo_img" alt="" src="static/img/logo.gif">
    <span class="wel_word">购物车</span>
    <%--  静态包含欢迎信息  --%>
    <%@ include file="/pages/common/success_menu.jsp" %>
</div>

<div id="main">

    <table>
        <tr>
            <td>商品名称</td>
            <td>数量</td>
            <td>单价</td>
            <td>金额</td>
            <td>操作</td>
        </tr>

        <c:if test="${empty sessionScope.cart.items}">
            <tr>
                <td colspan="5"><a href="index.jsp">亲,购物车空空如也!快去和小伙伴一起剁手吧!</a></td>
            </tr>
        </c:if>
        <c:if test="${not empty sessionScope.cart.items}">
            <c:forEach items="${sessionScope.cart.items}" var="entry">
                <tr>
                    <td>${entry.value.name}</td>
                    <td>
                        <input bookId="${entry.value.id}" type="text" style="width: 80px"
                               class="updateCount" value="${entry.value.count}">
```

```
                    </td>
                    <td>${entry.value.price}</td>
                    <td>${enrty.value.totalPrice}</td>
                    <td><a class="deleteItem" href="cartServlet?
action=deleteItem&id=${entry.value.id}">删除</a></td>
                </tr>
            </c:forEach>
        </c:if>


    </table>
    <c:if test="${not empty sessionScope.cart.items}">
        <div class="cart_info">
            <span class="cart_span">购物车中共有<span
class="b_count">${sessionScope.cart.totalCount}</span>件商品</span>
            <span class="cart_span">总金额<span
class="b_price">${sessionScope.cart.totalPrice}</span>元</span>
            <span class="cart_span"><a id="clearCart" href="cartServlet?
action=clear">清空购物车</a></span>
            <span class="cart_span"><a href="pages/cart/checkout.jsp">去结账
</a></span>
        </div>
    </c:if>
</div>

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

CartServlet.java

```
package com.admiral.web;

import com.admiral.pojo.Book;
import com.admiral.pojo.Cart;
import com.admiral.pojo.CartItem;
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;

/**
```

```
 * @author 白世鑫
 * @title: CartServlet
 * @projectName JavaWeb
 * @description:
 * @date 2020/9/8  2:48 上午
 */
public class CartServlet extends BaseServlet {

    private BookService bookService = new BookServiceImpl();


    protected void updateCount(HttpServletRequest req, HttpServletResponse
resp) throws ServletException, IOException {
        //获取请求的参数  商品编号  商品数量
        int id = WebUtils.parseInt(req.getParameter("id"),0);
        int count = WebUtils.parseInt(req.getParameter("count"),1);

        //获取购物车对象
        Cart cart = (Cart) req.getSession().getAttribute("cart");

        if (cart != null) {
            cart.updateCount(id,count);
            //从哪里来回哪里去
            resp.sendRedirect(req.getHeader("Referer"));
        }

    }
    protected void clear(HttpServletRequest req, HttpServletResponse resp)
throws ServletException, IOException {
        //1.先获取购物车对象
        Cart cart = (Cart) req.getSession().getAttribute("cart");
        if (cart != null) {
            cart.clear();
            //从哪里来回哪里去
            resp.sendRedirect(req.getHeader("Referer"));
        }

    }
    /**
     * 删除购物车商品项
     * @param req
     * @param resp
     * @throws ServletException
     * @throws IOException
     */
    protected void deleteItem(HttpServletRequest req, HttpServletResponse
resp) throws ServletException, IOException {
        int id = WebUtils.parseInt(req.getParameter("id"), 0);
```

```
        //先获取购物车对象
        Cart cart = (Cart) req.getSession().getAttribute("cart");

        if (cart != null) {
            cart.delteItem(id);
            //重定向会原来的页面
            resp.sendRedirect(req.getHeader("Referer"));
        }

    }


    protected void addItem(HttpServletRequest req, HttpServletResponse resp)
throws ServletException, IOException {
        //获取请求的参数,图书编号
        Integer id = WebUtils.parseInt(req.getParameter("id"), 0);
        //根据编号调动 service 查询图书信息
        Book book = bookService.queryBookById(id);
        //根据图书信息封装cartItem
        CartItem cartItem = new CartItem(book.getId(), book.getName(), 1,
book.getPrice(), book.getPrice());
        //调用 cart.additem 添加到购物车
        Cart cart = (Cart) req.getSession().getAttribute("cart");
        if (cart == null) {
            cart = new Cart();
            req.getSession().setAttribute("cart",cart);
        }
        cart.addItems(cartItem);
        System.out.println(cartItem);
        //重定向到首页
//        resp.sendRedirect(req.getContextPath());
        //重定向到原来的页面
        resp.sendRedirect(req.getHeader("Referer"));

    }
}
```

1.8 首页，购物车数据回显

在添加商品到购物车的时候，保存最后一个添加的商品名称

```
        //根据图书信息封装cartItem
        CartItem cartItem = new CartItem(book.getId(), book.getName(), count: 1, book.getPrice(), book.getPrice());
        //调用 cart.additem 添加到购物车
        Cart cart = (Cart) req.getSession().getAttribute( s: "cart");
        if (cart == null) {
            cart = new Cart();
            req.getSession().setAttribute( s: "cart",cart);
        }
        cart.addItems(cartItem);
        System.out.println(cartItem);

        //在添加商品的时候,保存最后一个添加商品的名称
        req.getSession().setAttribute( s: "lastName",cartItem.getName());

        //重定向到首页
//        resp.sendRedirect(req.getContextPath());
        //重定向到原来的页面
        resp.sendRedirect(req.getHeader( s: "Referer"));

    }
}
```

String s, **Object o**

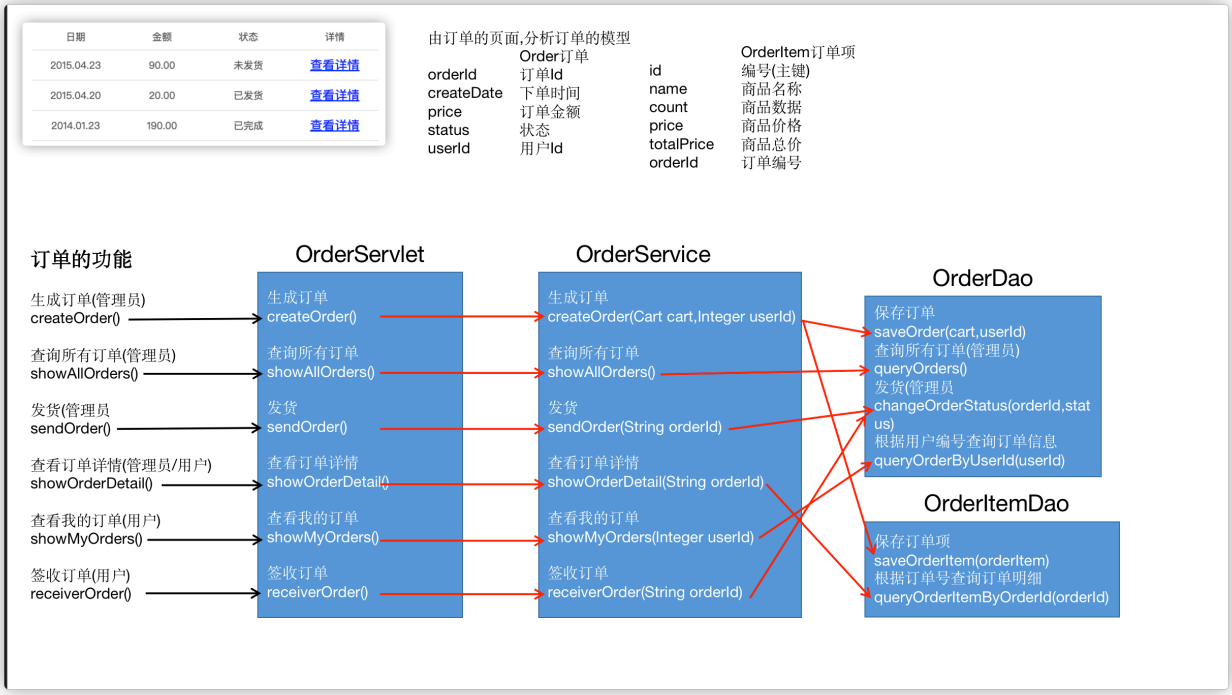
```
protected void addItem(HttpServletRequest req, HttpServletResponse resp)
throws ServletException, IOException {
        //获取请求的参数,图书编号
        Integer id = WebUtils.parseInt(req.getParameter("id"), 0);
        //根据编号调动 service 查询图书信息
        Book book = bookService.queryBookById(id);
        //根据图书信息封装cartItem
        CartItem cartItem = new CartItem(book.getId(), book.getName(), 1,
book.getPrice(), book.getPrice());
        //调用 cart.additem 添加到购物车
        Cart cart = (Cart) req.getSession().getAttribute("cart");
        if (cart == null) {
            cart = new Cart();
            req.getSession().setAttribute("cart",cart);
        }
        cart.addItems(cartItem);
        System.out.println(cartItem);

        //在添加商品的时候,保存最后一个添加商品的名称
        req.getSession().setAttribute("lastName",cartItem.getName());

        //重定向到首页
//        resp.sendRedirect(req.getContextPath());
        //重定向到原来的页面
        resp.sendRedirect(req.getHeader("Referer"));

    }
```

在 pages/client/index.jsp 页面中输出购物车信息


```
            <c:if test="${not empty sessionScope.cart.items}">
                <span>您的购物车中有${sessionScope.cart.totalCount}件商品</span>
                <div>
                    您刚刚将<span style="color: red">${sessionScope.lastName}
</span>加入到了购物车中
                </div>
            </c:if>
            <c:if test="${empty sessionScope.cart.items}">
                <div>
                    <span style="color: red">购物车为空</span>
                </div>
            </c:if>
```

2. 项目第七阶段：订单

2.1 订单模块的分析



2.2 订单模块的实现

2.2.1 创建订单模块的数据库表

t_order

```
create table t_order(
`order_id` varchar(50) primary key,
`create_time` datetime,
`price` decimal(11,2),
`status` int,
`user_id` int,
foreign key(`user_id`) references t_user(`id`)
);
```

t_order_item

```
create table t_order_item(
`id` int primary key auto_increment,
`name` varchar(100),
`count` int,
`price` decimal(11,2),
`total_price` decimal(11,2),
`order_id` varchar(50),
foreign key(`order_id`) references t_order(`order_id`)
);
```

2.2.2 创建订单模块的数据模型

Order.java

```
package com.admiral.pojo;

import java.math.BigDecimal;
import java.util.Date;

/**
 * @author 白世鑫
 * @title: Order
 * @projectName JavaWeb
 * @description:
 * @date 2020/9/8  11:42 下午
 */
public class Order {

    private String orderId;
    private Date createDate;
    private BigDecimal price;
    //0表示未发货,1表示已发货,2表示已签收
    private Integer status;
    private Integer userId;

    public Order() {
    }

    public Order(String orderId, Date createDate, BigDecimal price, Integer
status, Integer userId) {
        this.orderId = orderId;
        this.createDate = createDate;
        this.price = price;
        this.status = status;
        this.userId = userId;
    }

    public String getOrderId() {
        return orderId;
    }

    public void setOrderId(String orderId) {
        this.orderId = orderId;
    }

    public Date getCreateDate() {
        return createDate;
```

```
    }

    public void setCreateDate(Date createDate) {
        this.createDate = createDate;
    }

    public BigDecimal getPrice() {
        return price;
    }

    public void setPrice(BigDecimal price) {
        this.price = price;
    }

    public Integer getStatus() {
        return status;
    }

    public void setStatus(Integer status) {
        this.status = status;
    }

    public Integer getUserId() {
        return userId;
    }

    public void setUserId(Integer userId) {
        this.userId = userId;
    }

    @Override
    public String toString() {
        return "Order{" +
                "orderId='" + orderId + '\'' +
                ", createDate=" + createDate +
                ", price=" + price +
                ", status=" + status +
                ", userId=" + userId +
                '}';
    }
}
```

OrderItem.java

```
package com.admiral.pojo;
```

```java
import java.math.BigDecimal;

/**
 * @author 白世鑫
 * @title: OrderItem
 * @projectName JavaWeb
 * @description:
 * @date 2020/9/8  11:44 下午
 */
public class OrderItem {
    private Integer id;
    private String name;
    private Integer count;
    private BigDecimal price;
    private BigDecimal totalPrice;
    private String orderId;

    public OrderItem() {
    }

    public OrderItem(Integer id, String name, Integer count, BigDecimal price,
BigDecimal totalPrice, String orderId) {
        this.id = id;
        this.name = name;
        this.count = count;
        this.price = price;
        this.totalPrice = totalPrice;
        this.orderId = orderId;
    }

    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 Integer getCount() {
        return count;
    }
```

```
    public void setCount(Integer count) {
        this.count = count;
    }

    public BigDecimal getPrice() {
        return price;
    }

    public void setPrice(BigDecimal price) {
        this.price = price;
    }

    public BigDecimal getTotalPrice() {
        return totalPrice;
    }

    public void setTotalPrice(BigDecimal totalPrice) {
        this.totalPrice = totalPrice;
    }

    public String getOrderId() {
        return orderId;
    }

    public void setOrderId(String orderId) {
        this.orderId = orderId;
    }

    @Override
    public String toString() {
        return "OrderItem{" +
                "id=" + id +
                ", name='" + name + '\'' +
                ", count=" + count +
                ", price=" + price +
                ", totalPrice=" + totalPrice +
                ", orderId='" + orderId + '\'' +
                '}';
    }
}
```

2.2.3 编写订单模块的 Dao 程序和测试

OrderDao.java

```
package com.admiral.dao;

import com.admiral.pojo.Order;

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

    public int saveOrder(Order order);
}
```

OrderDaoImpl.java

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

import com.admiral.dao.BaseDao;
import com.admiral.dao.OrderDao;
import com.admiral.pojo.Order;

/**
 * @author 白世鑫
 * @title: OrderDaoImpl
 * @projectName JavaWeb
 * @description:
 * @date 2020/9/9  12:20 上午
 */
public class OrderDaoImpl extends BaseDao implements OrderDao {
    @Override
    public int saveOrder(Order order) {
        String sql = "insert into
t_order(order_id,create_time,price,status,user_id) values(?,?,?,?,?)";
        return
update(sql,order.getOrderId(),order.getCreateDate(),order.getPrice(),order.get
Status(),order.getUserId());
    }
}
```

OrderItemDao.java

```
package com.admiral.dao;

import com.admiral.pojo.OrderItem;

/**
 * @author 白世鑫
 * @description
 * @date 2020/9/9
 */
public interface OrderItemDao {
    public int saveOrderItem(OrderItem orderItem);
}
```

OrderItemDaoImpl.java

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

import com.admiral.dao.BaseDao;
import com.admiral.dao.OrderItemDao;
import com.admiral.pojo.OrderItem;

/**
 * @author 白世鑫
 * @title: OrderItemDaoImpl
 * @projectName JavaWeb
 * @description:
 * @date 2020/9/9  12:22 上午
 */
public class OrderItemDaoImpl extends BaseDao implements OrderItemDao {
    @Override
    public int saveOrderItem(OrderItem orderItem) {
        String sql = "insert into
t_order_item(name,count,price,total_price,order_id) values(?,?,?,?,?)";
        return
update(sql,orderItem.getName(),orderItem.getCount(),orderItem.getPrice(),order
Item.getTotalPrice(),orderItem.getOrderId());
    }
}
```

测试类OrderDaoImplTest:

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

import com.admiral.dao.OrderDao;
```

```
import com.admiral.dao.impl.OrderDaoImpl;
import com.admiral.pojo.Order;
import org.junit.Test;

import java.math.BigDecimal;
import java.util.Date;

/**
 * OrderDaoImpl Tester.
 *
 * @author <Authors name>
 * @version 1.0
 * @since <pre>9月 9, 2020</pre>
 */
public class OrderDaoImplTest {


    /**
     * Method: saveOrder(Order order)
     */
    @Test
    public void testSaveOrder() throws Exception {
        OrderDao orderDao = new OrderDaoImpl();
        orderDao.saveOrder(new Order("1234567890",new Date(),new
BigDecimal(100),0,1));
    }


}
```

OrderItemDaoImplTest.java

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

import com.admiral.dao.OrderItemDao;
import com.admiral.dao.impl.OrderItemDaoImpl;
import com.admiral.pojo.OrderItem;
import org.junit.Test;

import java.math.BigDecimal;

/**
 * OrderItemDaoImpl Tester.
 *
 * @author <Authors name>
 * @version 1.0
```

```
 * @since <pre>9月 9, 2020</pre>
 */
public class OrderItemDaoImplTest {


    /**
     * Method: saveOrderItem(OrderItem orderItem)
     */
    @Test
    public void testSaveOrderItem() throws Exception {
        OrderItemDao orderItemDao = new OrderItemDaoImpl();
        orderItemDao.saveOrderItem(new OrderItem(null,"钢铁是怎样炼成的",2,new
BigDecimal(100),new BigDecimal(200),"1234567890"));
        orderItemDao.saveOrderItem(new OrderItem(null,"那小子真帅",1,new
BigDecimal(300),new BigDecimal(300),"1234567890"));
        orderItemDao.saveOrderItem(new OrderItem(null,"金鳞岂是池中物",2,new
BigDecimal(500),new BigDecimal(1000),"1234567890"));
    }


}
```

2.2.4 编写订单模块的 Service 和测试

OrderService.java

```
package com.admiral.service;

import com.admiral.pojo.Cart;

/**
 * @author 白世鑫
 * @description
 * @date 2020/9/9
 */
public interface OrderService {
    public String createOrder(Cart cart, Integer userId);
}
```

OrderServiceImpl.java

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

```
import com.admiral.dao.OrderDao;
import com.admiral.dao.OrderItemDao;
import com.admiral.dao.impl.OrderDaoImpl;
import com.admiral.dao.impl.OrderItemDaoImpl;
import com.admiral.pojo.Cart;
import com.admiral.pojo.CartItem;
import com.admiral.pojo.Order;
import com.admiral.pojo.OrderItem;
import com.admiral.service.OrderService;

import java.util.Date;

/**
 * @author 白世鑫
 * @title: OrderServiceImpl
 * @projectName JavaWeb
 * @description:
 * @date 2020/9/9  12:38 上午
 */
public class OrderServiceImpl implements OrderService {
    @Override
    public String createOrder(Cart cart, Integer userId) {
        OrderDao orderDao = new OrderDaoImpl();
        OrderItemDao orderItemDao = new OrderItemDaoImpl();

        //先保存订单
        String orderId = System.currentTimeMillis() + "" + userId;
        Order order = new Order(orderId, new Date(), cart.getTotalPrice(), 0,
userId);
        orderDao.saveOrder(order);

        //再保存订单项
        //遍历购物车,将购物车商品项转换为订单项
        for (CartItem cartItem : cart.getItems().values()) {
            OrderItem orderItem = new
OrderItem(null,cartItem.getName(),cartItem.getCount(),cartItem.getPrice(),cart
Item.getTotalPrice(),orderId);
            orderItemDao.saveOrderItem(orderItem);
        }

        //清空购物车
        return orderId;
    }
}
```

OrderServiceImplTest.java

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

```
import com.admiral.pojo.Cart;
import com.admiral.pojo.CartItem;
import com.admiral.service.OrderService;
import com.admiral.service.impl.OrderServiceImpl;
import org.junit.Test;

import java.math.BigDecimal;

/**
 * OrderServiceImpl Tester.
 *
 * @author <Authors name>
 * @version 1.0
 * @since <pre>9月 9, 2020</pre>
 */
public class OrderServiceImplTest {

    /**
     * Method: createOrder(Cart cart, Integer userId)
     */
    @Test
    public void testCreateOrder() throws Exception {
        Cart cart = new Cart();

        cart.addItems(new CartItem(1, "时间简史", 1, new BigDecimal(100), new
BigDecimal(100)));
        cart.addItems(new CartItem(1, "时间简史", 1, new BigDecimal(100), new
BigDecimal(100)));
        cart.addItems(new CartItem(2, "钢铁是怎样炼成的", 1, new BigDecimal(200),
new BigDecimal(200)));

        OrderService orderService = new OrderServiceImpl();
        System.out.println("订单号为:" + orderService.createOrder(cart, 1));
    }


}
```

2.2.5 编写订单模块的 web 层和页面联调

OrderServlet.java

```
package com.admiral.web;

import com.admiral.pojo.Cart;
```

```
import com.admiral.pojo.User;
import com.admiral.service.OrderService;
import com.admiral.service.impl.OrderServiceImpl;

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

/**
 * @author 白世鑫
 * @title: OrderServlet
 * @projectName JavaWeb
 * @description:
 * @date 2020/9/9  12:57 上午
 */
public class OrderServlet extends BaseServlet{

    OrderService orderService = new OrderServiceImpl();

    protected void createOrder(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        //先获取购物车对象
        Cart cart = (Cart) request.getSession().getAttribute("cart");

        //获取登陆的User对象
        User login = (User) request.getSession().getAttribute("user");
        if (login == null) {

 request.getRequestDispatcher("pages/user/login.jsp").forward(request,response
);
            return;
        }
        //获取 userId
        Integer userId = login.getId();

        //调用 service 生成订单
        String orderId = orderService.createOrder(cart, userId);
        //保存订单号
        request.setAttribute("orderId",orderId);

        //跳转页面

 request.getRequestDispatcher("pages/cart/checkout.jsp").forward(request,respo
nse);
    }
}
```

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd"
         version="4.0">

    <servlet>
        <servlet-name>OrderServlet</servlet-name>
        <servlet-class>com.admiral.web.OrderServlet</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>OrderServlet</servlet-name>
        <url-pattern>/orderServlet</url-pattern>
    </servlet-mapping>

</web-app>
```


```
<span class="cart_span"><a href="orderServlet?action=createOrder">去结账</a>
</span>
```

跳转页面数据展示

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>结算页面</title>
  <%--  静态包含 base css样式 jQuery  --%>
  <%@ include file="/pages/common/header.jsp"%>
<style type="text/css">
  h1 {
    text-align: center;
    margin-top: 200px;
  }
```

```
</style>
</head>
<body>

  <div id="header">
      <img class="logo_img" alt="" src="../../static/img/logo.gif" >
      <span class="wel_word">结算</span>
    <%--  静态包含欢迎信息  --%>
    <%@ include file="/pages/common/success_menu.jsp"%>
  </div>

  <div id="main">

    <h1>你的订单已结算，订单号为${requestScope.orderId}</h1>


  </div>

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

2.2.6 Bug

修改 OrderService 中的代码,更新库存和销量

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

import com.admiral.dao.BookDao;
import com.admiral.dao.OrderDao;
import com.admiral.dao.OrderItemDao;
import com.admiral.dao.impl.BookDaoImpl;
import com.admiral.dao.impl.OrderDaoImpl;
import com.admiral.dao.impl.OrderItemDaoImpl;
import com.admiral.pojo.*;
import com.admiral.service.OrderService;

import java.util.Date;

/**
 * @author 白世鑫
 * @title: OrderServiceImpl
 * @projectName JavaWeb
 * @description:
```

```
 * @date 2020/9/9  12:38 上午
 */
public class OrderServiceImpl implements OrderService {
    private OrderDao orderDao = new OrderDaoImpl();
    private OrderItemDao orderItemDao = new OrderItemDaoImpl();
    private BookDao bookDao = new BookDaoImpl();

    @Override
    public String createOrder(Cart cart, Integer userId) {
        //先保存订单
        String orderId = System.currentTimeMillis() + "" + userId;
        Order order = new Order(orderId, new Date(), cart.getTotalPrice(), 0,
userId);
        orderDao.saveOrder(order);

        //再保存订单项
        //遍历购物车,将购物车商品项转换为订单项
        for (CartItem cartItem : cart.getItems().values()) {
            OrderItem orderItem = new
OrderItem(null,cartItem.getName(),cartItem.getCount(),cartItem.getPrice(),cart
Item.getTotalPrice(),orderId);
            //保存订单项到数据库
            orderItemDao.saveOrderItem(orderItem);

            //更新库存和销量
            Book book = bookDao.queryBookById(cartItem.getId());
            book.setSales(book.getSales() + cartItem.getCount());
            book.setStock(book.getStock() - cartItem.getCount());
            bookDao.updateBook(book);
        }

        //清空购物车
        cart.clear();
        return orderId;
    }
}
```

OrderServlet.java

```
package com.admiral.web;

import com.admiral.pojo.Cart;
import com.admiral.pojo.User;
import com.admiral.service.OrderService;
import com.admiral.service.impl.OrderServiceImpl;
```

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

/**
 * @author 白世鑫
 * @title: OrderServlet
 * @projectName JavaWeb
 * @description:
 * @date 2020/9/9  12:57 上午
 */
public class OrderServlet extends BaseServlet{

    OrderService orderService = new OrderServiceImpl();

    protected void createOrder(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        //先获取购物车对象
        Cart cart = (Cart) request.getSession().getAttribute("cart");

        //获取登陆的User对象
        User login = (User) request.getSession().getAttribute("user");
        if (login == null) {

 request.getRequestDispatcher("pages/user/login.jsp").forward(request,response
);
            return;
        }
        //获取 userId
        Integer userId = login.getId();

        //调用 service 生成订单
        String orderId = orderService.createOrder(cart, userId);
        //保存订单号
//        request.setAttribute("orderId",orderId);
        request.getSession().setAttribute("orderId",orderId);

        //跳转页面
//
request.getRequestDispatcher("pages/cart/checkout.jsp").forward(request,respon
se);
        response.sendRedirect(request.getContextPath() +
"/pages/cart/checkout.jsp");
    }
}
```

checkout.jsp

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>结算页面</title>
  <%--  静态包含 base css样式 jQuery  --%>
  <%@ include file="/pages/common/header.jsp"%>
<style type="text/css">
  h1 {
    text-align: center;
    margin-top: 200px;
  }
</style>
</head>
<body>

  <div id="header">
      <img class="logo_img" alt="" src="../../static/img/logo.gif" >
      <span class="wel_word">结算</span>
    <%--  静态包含欢迎信息  --%>
    <%@ include file="/pages/common/success_menu.jsp"%>
  </div>

  <div id="main">

    <h1>你的订单已结算，订单号为${sessionScope.orderId}</h1>


  </div>

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