

# 13장.

## 쇼핑몰 - 주문 처리와 게시판

이번 장에서는 상품에 대한 고객의 주문 과정과 Q&A 1:1 게시판 작성하는 방법을 학습합니다.

13-1 전반적인 흐름 분석하기

13-2 장바구니 담기

13-3 주문 처리하기

13-4 Q&A 게시판

## 13-1 전반적인 흐름 분석하기

Nonage Shop은 비회원인 경우 장바구니, 주문을 사용할 수 없도록 할 것입니다. 주문 결제 방식도 무통장 입금으로 할 것이며 앞으로 배울 관리자가 입금을 확인을 하는 방식으로 처리됩니다.

장바구니는 쿠키나 세션으로 처리하는 방법이 있지만 쿠키는 보안상의 문제점이 있을 수 있고 세션은 메모리에 부담을 주기 때문에 Nonage Shop은 데이터베이스로 처리하는 방식을 취하겠습니다. 일단 장바구니를 선택하면 장바구니(cart) 테이블에 저장하고 고객이 주문을 하면 cart 테이블에 있는 데이터가 주문 테이블에 저장되고 cart 테이블의 데이터를 삭제합니다.

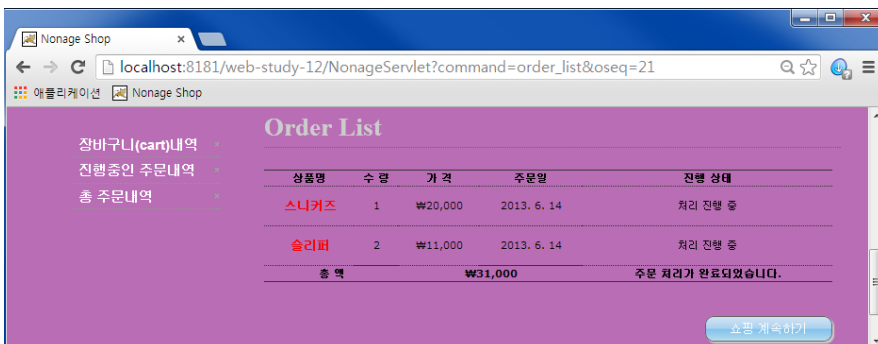
### ▼ 장바구니 확인 화면

로그인 후 상품 상세 화면에서 장바구니에 상품을 담게 되면 다음과 같이 장바구니 확인 화면으로 이동합니다. 장바구니에 담기 상품의 수량과 가격을 확인하는 한 후에 쇼핑을 계속하거나 장바구니에 담긴 상품을 주문할 수 있게 됩니다. 「주문하기」버튼을 클릭하면 장바구니에 들어있는 상품이 주문 처리되고 진행 중인 주문 리스트 화면으로 이동한다.



### ▼ 주문 확인 화면

주문 처리가 완료되었음을 표시하는 화면입니다.



## 13-2 장바구니 담기

장바구니와 주문 처리를 위해 관련된 부분을 진행해 보도록 하겠습니다. 장바구니에 상품을 담기 위해서는 데이터베이스의 cart 테이블에 정보를 저장해야 합니다. 그러기 위해서는 cart 테이블을 처리하기 위한 DAO를 작성해야 합니다. 장바구니에 상품을 담기 위한 DAO를 만들기 위해서 장바구니 정보를 담을 VO부터 만듭시다.

[직접해보세요] 이클립스에서 장바구니 정보를 저장하는 VO 클래스 만들기

[파일 이름 : comWnonageWdtoWCartVO.java]

```
1 package com.nonage.dto;
2
3 import java.sql.Timestamp;
4
5 public class CartVO {
6     private int cseq;
7     private String id;
8     private int pseq;
9     private String mname;
10    private String pname;
11    private int quantity;
12    private int price2;
13    private Timestamp indate;
14
15
16    public int getPrice2() {
17        return price2;
18    }
19
20    public void setPrice2(int price2) {
21        this.price2 = price2;
22    }
23
24    public int getCseq() {
25        return cseq;
26    }
27
28    public void setCseq(int cseq) {
29        this.cseq = cseq;
30    }
31
32    public String getId() {
33        return id;
34    }
35
36    public void setId(String id) {
37        this.id = id;
38    }
39 }
```

```

38 public int getPseq() {
39     return pseq;
40 }
41 public void setPseq(int pseq) {
42     this.pseq = pseq;
43 }
44
45 public String getMname() {
46     return mname;
47 }
48 public void setMname(String mname) {
49     this.mname = mname;
50 }
51 public String getPname() {
52     return pname;
53 }
54
55 public void setPname(String pname) {
56     this.pname = pname;
57 }
58 public int getQuantity() {
59     return quantity;
60 }
61 public void setQuantity(int quantity) {
62     this.quantity = quantity;
63 }
64
65 public Timestamp getIndate() {
66     return indate;
67 }
68 public void setIndate(Timestamp indate) {
69     this.indate = indate;
70 }
71 }
72 }

```

장바구니 관리를 위한 CartDAO를 만듭시다.

**[직접해보세요]** 장바구니 테이블을 액세스하는 DAO 클래스 만들기

[파일 이름 :srcWcomWnonageWdaoWCartDAO.java]

```

1 package com.nonage.dao;
2
3 import java.sql.Connection;
4 import java.sql.PreparedStatement;
5 import java.sql.ResultSet;
6 import java.util.ArrayList;
7
8 import util.DBManager;

```

```

9
10 import com.nonage.dto.CartVO;
11
12 public class CartDAO {
13
14     private CartDAO() {
15     }
16
17     private static CartDAO instance = new CartDAO();
18
19     public static CartDAO getInstance() {
20         return instance;
21     }
22
23     public void insertCart(CartVO cartVO) {
24         String sql = "insert into cart(cseq,id, pseq, quantity)" +
25             " values(cart_seq.nextval,?, ?, ?)";
26
27         Connection conn = null;
28         PreparedStatement pstmt = null;
29
30         try {
31             conn = DBManager.getConnection();
32             pstmt = conn.prepareStatement(sql);
33             pstmt.setString(1, cartVO.getId());
34             pstmt.setInt(2, cartVO.getPseq());
35             pstmt.setInt(3, cartVO.getQuantity());
36             pstmt.executeUpdate();
37         } catch (Exception e) {
38             e.printStackTrace();
39         } finally {
40             DBManager.close(conn, pstmt);
41         }
42     }
43
44     public ArrayList<CartVO> listCart(String userId) {
45         ArrayList<CartVO> cartList = new ArrayList<CartVO>();
46         String sql = "select * from cart_view where id=? order by cseq desc";
47
48         Connection conn = null;
49         PreparedStatement pstmt = null;
50         ResultSet rs = null;
51
52         try {
53             conn = DBManager.getConnection();
54             pstmt = conn.prepareStatement(sql);
55             pstmt.setString(1, userId);
56             rs = pstmt.executeQuery();

```

```

57     while (rs.next()) {
58         CartVO cartVO = new CartVO();
59         cartVO.setCseq(rs.getInt(1));
60         cartVO.setId(rs.getString(2));
61         cartVO.setPseq(rs.getInt(3));
62         cartVO.setMname(rs.getString(4));
63         cartVO.setPname(rs.getString(5));
64         cartVO.setQuantity(rs.getInt(6));
65         cartVO.setIndate(rs.getTimestamp(7));
66         cartVO.setPrice2(rs.getInt(8));
67         cartList.add(cartVO);
68     }
69 } catch (Exception e) {
70     e.printStackTrace();
71 }finally {
72     DBManager.close(conn, pstmt, rs);
73 }
74 return cartList;
75 }
76
77 public void deleteCart(int cseq) {
78     String sql = "delete cart where cseq=?";
79
80     Connection conn = null;
81     PreparedStatement pstmt = null;
82
83     try {
84         conn = DBManager.getConnection();
85         pstmt = conn.prepareStatement(sql);
86         pstmt.setInt(1, cseq);
87         pstmt.executeUpdate();
88     } catch (Exception e) {
89         e.printStackTrace();
90     } finally {
91         DBManager.close(conn, pstmt);
92     }
93 }
94 }

```

주문 처리를 하는 페이지들을 위한 mypage 폴더를 만듭니다. 이 폴더에 공통적으로 들어가는 파일들을 살펴보도록 합시다.

**[직접해보세요]** 서브 페이지에서 이미지를 출력하는 파일

[파일이름 : WebContent/mypage/sub\_img.html]

1	<div id="sub_img">
2	

3	</div>
4	<div class="clear"></div>

**[직접해보세요]** 서브 페이지에서 좌측 메뉴를 구성하는 파일

[파일이름 : WebContent/mypage/sub\_menu.jsp]

1	<%@ page language="java" contentType="text/html; charset=UTF-8"
2	pageEncoding="UTF-8"%>
3	<meta charset="UTF-8">
4	<nav id="sub_menu">
5	<ul>
6	<li><a href="NonageServlet?command=cart_list">장바구니(cart)내역</a></li>
7	<li><a href="NonageServlet?command=mypage">진행중인 주문내역</a></li>
8	<li><a href="NonageServlet?command=order_all">총 주문내역</a></li>
9	</ul>
10	</nav>

주문 처리를 위한 폼에 입력된 정보가 올바른지 판단하는 자바스크립트 파일을 WebContent의 하위 폴더인 mypage에 mypage.js란 이름으로 작성합니다.

자바스크립트 파일은 이미 header.jsp 페이지에서 포함하고 있습니다.

header.jsp	7<head>
8	<meta charset="UTF-8">
9	<title>Nonage Shop</title>
10	<link href="css/shopping.css" rel="stylesheet">
11	<script type="text/javascript" src="member/member.js"></script>
12	<script type="text/javascript" src="mypage/mypage.js"></script>
13	</head>

**[직접해보세요]** 폼에 입력된 정보가 올바른지 판단하는 자바스크립트 파일 [파일이름 :

WebContent/mypage/mypage.js]

1	function go_cart() {
2	if (document.formm.quantity.value == "") {
3	alert("수량을 입력하여 주세요.");
4	document.formm.quantity.focus();
5	} else {
6	document.formm.action = "NonageServlet?command=cart_insert";
7	document.formm.submit();
8	}
9	}
10	
11	function go_cart_delete() {
12	var count = 0;
13	
14	if (document.formm.cseq.length == undefined) {
15	if (document.formm.cseq.checked == true) {

```

16     count++;
17 }
18 }
19
20 for ( var i = 0; i < document.formm.cseq.length; i++) {
21     alert("" + document.formm.cseq[i].checked);
22     if (document.formm.cseq[i].checked == true) {
23         count++;
24         alert("" + count);
25     }
26 }
27 if (count == 0) {
28     alert("삭제할 항목을 선택해 주세요.");
29
30 } else {
31     document.formm.action = "NonageServlet?command=cart_delete";
32     document.formm.submit();
33 }
34 }
35
36 function go_order_insert() {
37     document.formm.action = "NonageServlet?command=order_insert";
38     document.formm.submit();
39 }
40
41 function go_order_delete() {
42     var count = 0;
43
44     if (document.formm.oseq.length == undefined) {
45         if (document.formm.oseq.checked == true) {
46             count++;
47         }
48     }
49
50     for ( var i = 0; i < document.formm.oseq.length; i++) {
51         if (document.formm.oseq[i].checked == true) {
52             count++;
53         }
54     }
55     if (count == 0) {
56         alert("삭제할 항목을 선택해 주세요.");
57
58     } else {
59         document.formm.action = "NonageServlet?command=order_delete";
60         document.formm.submit();
61     }
62 }
63

```



```

64 function go_order() {
65     document.formm.action = "NonageServlet?command=mypage";
66     document.formm.submit();
67 }

```

상품 상세 보기 페이지에서 [장바구니에 담기] 버튼을 클릭하면 이를 처리하기 위한 “http://localhost:8181/web-study-12/NonageServlet?command=cart\_insert” 요청이 발생합니다. 이 요청을 받아 처리하는 액션을 만들어 봅시다.

**[직접해보세요]** 장바구니에 상품을 담기 위한 액션 클래스

[파일이름 : srcWcomWnonageWcontrollerWactionWCartInsertAction.java]

```

1  package com.nonage.controller.action;
2
3  import java.io.IOException;
4
5  import javax.servlet.ServletException;
6  import javax.servlet.http.HttpServletRequest;
7  import javax.servlet.http.HttpServletResponse;
8  import javax.servlet.http.HttpSession;
9
10 import com.nonage.dao.CartDAO;
11 import com.nonage.dto.CartVO;
12 import com.nonage.dto.MemberVO;
13
14 public class CartInsertAction implements Action {
15
16     @Override
17     public void execute(HttpServletRequest request, HttpServletResponse response)
18         throws ServletException, IOException {
19         String url = "NonageServlet?command=cart_list";
20
21         HttpSession session = request.getSession();
22         MemberVO loginUser = (MemberVO) session.getAttribute("loginUser");
23         if (loginUser == null) {
24             url = "NonageServlet?command=login_form";
25         } else {
26             CartVO cartVO = new CartVO();
27             cartVO.setIdx(loginUser.getId());
28             cartVO.setPseq(Integer.parseInt(request.getParameter("pseq")));
29             cartVO.setQuantity(Integer.parseInt(request
30                 .getParameter("quantity")));
31
32             CartDAO cartDAO = CartDAO.getInstance();
33             cartDAO.insertCart(cartVO);
34         }
35         response.sendRedirect(url);

```

```

36     }
37     }

```

**[직접해보세요]** 커맨드(command) 패턴으로 작업 처리를 위한 명령 처리 클래스  
 ActionFactory 수정하기 [파일 이름 : srcWcomWnonageWcontrollerWActionFactory.java]

```

..    //중복되는 내용 생략
40        } else if (command.equals("cart_insert")) {
41            action = new CartInsertAction();
42        }
..    //중복되는 내용 생략

```

장바구니에 상품을 담고 나면 "NonageServlet?command=cart\_list" 요청이 발생합니다. 이  
 요청은 장바구니에 현재 담겨진 장바구니 리스트를 출력합니다. 이 요청을 처리할 액션 클  
 래스를 만듭니다.

**[직접해보세요]** 장바구니 리스트를 처리한 액션 클래스  
 [파일 이름 : srcWcomWnonageWcontrollerWactionWCartListAction.java]

```

1  package com.nonage.controller.action;
2
3  import java.io.IOException;
4  import java.util.ArrayList;
5
6  import javax.servlet.ServletException;
7  import javax.servlet.http.HttpServletRequest;
8  import javax.servlet.http.HttpServletResponse;
9  import javax.servlet.http.HttpSession;
10
11 import com.nonage.dao.CartDAO;
12 import com.nonage.dto.CartVO;
13 import com.nonage.dto.MemberVO;
14
15 public class CartListAction implements Action {
16
17     @Override
18     public void execute(HttpServletRequest request, HttpServletResponse response)
19         throws ServletException, IOException {
20         String url = "mypage/cartList.jsp";
21
22         HttpSession session = request.getSession();
23         MemberVO loginUser = (MemberVO) session.getAttribute("loginUser");
24         if (loginUser == null) {
25             url = "NonageServlet?command=login_form";
26         } else {
27             CartDAO cartDAO = CartDAO.getInstance();
28             ArrayList<CartVO> cartList = cartDAO.listCart(loginUser.getId());

```

```

29
30     int totalPrice = 0;
31     for (CartVO cartVO : cartList) {
32         totalPrice += cartVO.getPrice2() * cartVO.getQuantity();
33     }
34
35     request.setAttribute("cartList", cartList);
36     request.setAttribute("totalPrice", totalPrice);
37 }
38 request.getRequestDispatcher(url).forward(request, response);
39 }
40 }

```

**[직접해보세요]** 장바구니 리스트를 출력하는 JSP 파일

[파일이름 : WebContent/mypage/cartList.jsp]

```

1  <%@ page language="java" contentType="text/html; charset=EUC-KR"
2      pageEncoding="EUC-KR"%>
3
4  <%@ include file="../header.jsp" %>
5  <%@ include file="sub_img.html"%>
6  <%@ include file="sub_menu.jsp" %>
7  <article>
8      <h2> Cart List </h2>
9      <form name="formm" method="post">
10         <c:choose>
11             <c:when test= "${cartList.size() == 0}">
12                 <h3 style="color: red;text-align: center;"> 장바구니가 비었습니다. </h3>
13             </c:when>
14             <c:otherwise>
15                 <table id="cartList">
16                     <tr>
17                         <th>상품명</th><th>수량</th><th>가격</th><th>주문일</th><th>삭제</th>
18                     </tr>
19
20                     <c:forEach items="${cartList}" var="cartVO">
21                         <tr>
22                             <td>
23                                 <a href=
24 "NonageServlet?command=product_detail&pseq=${cartVO.pseq}">
25                                     <h3> ${cartVO.pname} </h3>
26                                 </a>
27                             </td>
28                             <td> ${cartVO.quantity} </td>
29                             <td>
30                                 <fmt:formatNumber value="${cartVO.price2*cartVO.quantity}"
31                                     type="currency"/>
32                             </td>

```

```

33         <td><fmt:formatDate value="${cartVO.indate}" type="date"/></td>
34         <td><input type="checkbox" name="cseq" value= "${cartVO.cseq}"> </td>
35     </tr>
36 </c:forEach>
37
38     <tr>
39         <th colspan="2"> 총 액 </th>
40         <th colspan="2">
41             <fmt:formatNumber value="$ {totalPrice}" type="currency"/><br>
42         </th>
43         <th><a href="#" onclick="go_cart_delete()"><h3>삭제하기</h3></a></th>
44     </tr>
45 </table>
46 </c:otherwise>
47 </c:choose>
48
49 <div class="clear"></div>
50
51 <div id="buttons" style="float: right">
52     <input type="button" value="쇼핑 계속하기" class="cancel"
53     onclick="location.href='NonageServlet?command=index'">
54     <c:if test= "${cartList.size() != 0}">
55         <input type="button" value="주문하기" class="submit"
56         onclick="go_order_insert()">
57     </c:if>
58 </div>
59 </form>
60 </article>
61 <%@ include file="../footer.jsp" %>

```

**[직접해보세요]** 커맨드(command) 패턴으로 작업 처리를 위한 명령 처리 클래스  
 ActionFactory 수정하기 [파일 이름 : srcWcomWnonageWcontrollerWActionFactory.java]

```

.. //중복되는 내용 생략
42     } else if (command.equals("cart_list")) {
43         action = new CartListAction();
44     }
.. //중복되는 내용 생략

```

장바구니에 담긴 상품 중에서 맘이 바뀌어서 주문하지 않을 상품은 장바구니에서 삭제할 수 있습니다. 삭제하고자 하는 품목에 대해 체크 박스를 체크한 후 [삭제하기]를 클릭하면 "NonageServlet?command=cart\_delete" 요청이 발생합니다. 이 요청을 받아 처리하는 액션을 만들어 봅시다.

**[직접해보세요]** 장바구니에 상품을 삭제하기 위한 액션 클래스  
 [파일 이름 : srcWcomWnonageWcontrollerWactionWCartDeleteAction.java]

```

1 package com.nonage.controller.action;
2
3 import java.io.IOException;
4
5 import javax.servlet.ServletException;
6 import javax.servlet.http.HttpServletRequest;
7 import javax.servlet.http.HttpServletResponse;
8
9 import com.nonage.dao.CartDAO;
10
11 public class CartDeleteAction implements Action {
12
13     @Override
14     public void execute(HttpServletRequest request, HttpServletResponse response)
15         throws ServletException, IOException {
16         String url = "NonageServlet?command=cart_list";
17
18         String[] cseqArr = request.getParameterValues("cseq");
19
20         for(String cseq:cseqArr){
21             System.out.println(cseq);
22             CartDAO cartDAO = CartDAO.getInstance();
23             cartDAO.deleteCart(Integer.parseInt(cseq));
24         }
25         request.getRequestDispatcher(url).forward(request, response);
26     }
27 }

```

**[직접해보세요]** 커맨드(command) 패턴으로 작업 처리를 위한 명령 처리 클래스  
ActionFactory 수정하기 [파일 이름 : src\com\Wnonage\Wcontroller\WActionFactory.java]

```

.. //중복되는 내용 생략
44     } else if (command.equals("cart_delete")) {
45         action = new CartDeleteAction();
46     }
.. //중복되는 내용 생략

```

## 13-3 주문 처리하기

이번에는 주문 처리를 해보도록 하겠습니다. 주문을 위해서는 데이터베이스의 order 테이블에 정보를 저장해야 합니다. 그러기 위해서는 order 테이블을 처리하기 위한 DAO를 작성해야 합니다. 주문 처리를 위한 DAO를 만들기 위해서 주문 정보를 담을 VO부터 만듭시다.

[직접해보세요] 이클립스에서 주문 정보를 저장하는 VO 클래스 만들기

[파일 이름 : comWnonageWdtoWOrderVO.java]

```
1 package com.nonage.dto;
2
3 import java.sql.Timestamp;
4
5 public class OrderVO {
6     private int odseq;
7     private int oseq;
8     private String id;
9     private Timestamp indate;
10    private String mname;
11    private String zipNum;
12    private String address;
13    private String phone;
14    private int pseq;
15    private String pname;
16    private int quantity;
17    private int price2;
18    private String result;
19
20    public int getOdseq() {
21        return odseq;
22    }
23    public void setOdseq(int odseq) {
24        this.odseq = odseq;
25    }
26    public int getPrice2() {
27        return price2;
28    }
29    public void setPrice2(int price2) {
30        this.price2 = price2;
31    }
32    public int getOseq() {
33        return oseq;
34    }
35    public void setOseq(int oseq) {
36        this.oseq = oseq;
37    }
38    public String getId() {
39        return id;
```

```

40     }
41     public void setId(String id) {
42         this.id = id;
43     }
44     public int getPseq() {
45         return pseq;
46     }
47     public void setPseq(int pseq) {
48         this.pseq = pseq;
49     }
50     public String getMname() {
51         return mname;
52     }
53     public void setMname(String mname) {
54         this.mname = mname;
55     }
56     public String getPname() {
57         return pname;
58     }
59     public void setPname(String pname) {
60         this.pname = pname;
61     }
62     public int getQuantity() {
63         return quantity;
64     }
65     public void setQuantity(int quantity) {
66         this.quantity = quantity;
67     }
68     public String getResult() {
69         return result;
70     }
71     public void setResult(String result) {
72         this.result = result;
73     }
74     public String getZipNum() {
75         return zipNum;
76     }
77     public void setZipNum(String zipNum) {
78         this.zipNum = zipNum;
79     }
80     public String getAddress() {
81         return address;
82     }
83     public void setAddress(String address) {
84         this.address = address;
85     }
86     public String getPhone() {
87         return phone;

```

```

88     }
89     public void setPhone(String phone) {
90         this.phone = phone;
91     }
92     public Timestamp getIndate() {
93         return indate;
94     }
95     public void setIndate(Timestamp indate) {
96         this.indate = indate;
97     }
98 }

```

주문 처리를 위한 OrderDAO를 만듭시다.

**[직접해보세요]** 주문 테이블을 액세스하는 DAO 클래스 만들기

[파일 이름 :srcWcomWnonageWdaoWOrderDAO.java]

```

1  package com.nonage.dao;
2
3  import java.sql.Connection;
4  import java.sql.PreparedStatement;
5  import java.sql.ResultSet;
6  import java.util.ArrayList;
7
8  import util.DBManager;
9
10 import com.nonage.dto.CartVO;
11 import com.nonage.dto.OrderVO;
12
13 public class OrderDAO {
14
15     private OrderDAO() {
16     }
17
18     private static OrderDAO instance = new OrderDAO();
19
20     public static OrderDAO getInstance() {
21         return instance;
22     }
23
24     public int insertOrder(ArrayList<CartVO> cartList, String id) {
25         int maxOseq = 0;
26
27         Connection conn = null;
28         PreparedStatement pstmt = null;
29         ResultSet rs;
30
31         try {
32             conn = DBManager.getConnection();

```



```

33
34     String selectMaxOseq = "select max(oseq) from orders";
35     pstmt = conn.prepareStatement(selectMaxOseq);
36     rs = pstmt.executeQuery();
37     if (rs.next()) {
38         maxOseq = rs.getInt(1);
39     }
40     pstmt.close();
41
42     String insertOrder = "insert into orders(oseq, id) values(" +
43         "orders_seq.nextval, ?)";
44     pstmt = conn.prepareStatement(insertOrder);
45     pstmt.setString(1, id);
46     pstmt.executeUpdate();
47
48     for (CartVO cartVO : cartList) {
49         insertOrderDetail(cartVO, maxOseq);
50     }
51 } catch (Exception e) {
52     e.printStackTrace();
53 } finally {
54     DBManager.close(conn, pstmt);
55 }
56 return maxOseq;
57 }
58
59 public void insertOrderDetail(CartVO cartVO, int maxOseq) {
60     Connection conn = null;
61     PreparedStatement pstmt = null;
62
63     try {
64         conn = DBManager.getConnection();
65
66         String insertOrderDetail = "insert into order_detail(odseq, oseq, " +
67             "pseq, quantity) values(order_detail_seq.nextval, ?, ?, ?)";
68         pstmt = conn.prepareStatement(insertOrderDetail);
69         pstmt.setInt(1, maxOseq);
70         pstmt.setInt(2, cartVO.getPseq());
71         pstmt.setInt(3, cartVO.getQuantity());
72         pstmt.executeUpdate();
73         pstmt.close();
74
75         String updateCartResult = "update cart set result=2 where cseq=?";
76         pstmt = conn.prepareStatement(updateCartResult);
77         pstmt.setInt(1, cartVO.getCseq());
78         pstmt.executeUpdate();
79     } catch (Exception e) {
80         e.printStackTrace();

```

```

81     } finally {
82         DBManager.close(conn, pstmt);
83     }
84 }
85
86 public ArrayList<OrderVO> listOrderById(String id, String result, int oseq) {
87     ArrayList<OrderVO> orderList = new ArrayList<OrderVO>();
88     String sql = "select * from order_view where id=? "
89         + "and result like '%||?||%' and oseq=?";
90     Connection conn = null;
91     PreparedStatement pstmt = null;
92     ResultSet rs = null;
93     try {
94         conn = DBManager.getConnection();
95         pstmt = conn.prepareStatement(sql);
96         pstmt.setString(1, id);
97         pstmt.setString(2, result);
98         pstmt.setInt(3, oseq);
99         rs = pstmt.executeQuery();
100        while (rs.next()) {
101            OrderVO orderVO = new OrderVO();
102            orderVO.setOdseq(rs.getInt("ODSEQ"));
103            orderVO.setOseq(rs.getInt("OSEQ"));
104            orderVO.setId(rs.getString("ID"));
105            orderVO.setIndate(rs.getTimestamp("INDATE"));
106            orderVO.setMname(rs.getString("MNAME"));
107            orderVO.setZipNum(rs.getString("ZIP_NUM"));
108            orderVO.setAddress(rs.getString("ADDRESS"));
109            orderVO.setPhone(rs.getString("PHONE"));
110            orderVO.setPseq(rs.getInt("PSEQ"));
111            orderVO.setQuantity(rs.getInt("QUANTITY"));
112            orderVO.setPname(rs.getString("PNAME"));
113            orderVO.setPrice2(rs.getInt("PRICE2"));
114            orderVO.setResult(rs.getString("RESULT"));
115            orderList.add(orderVO);
116        }
117    } catch (Exception e) {
118        e.printStackTrace();
119    } finally {
120        DBManager.close(conn, pstmt, rs);
121    }
122    return orderList;
123 }
124
125 public ArrayList<Integer> selectSeqOrderIng(String id) {
126     ArrayList<Integer> oseqList = new ArrayList<Integer>();
127     String sql = "select distinct oseq from order_view " +
128         "where id=? and result='1' order by oseq desc";

```

```

129
130     Connection con = null;
131     PreparedStatement pstmt = null;
132     ResultSet rs = null;
133
134     try {
135         con = DBManager.getConnection();
136         pstmt = con.prepareStatement(sql);
137         pstmt.setString(1, id);
138         rs = pstmt.executeQuery();
139         while (rs.next()) {
140             oseqList.add(rs.getInt(1));
141         }
142     } catch (Exception e) {
143         e.printStackTrace();
144     } finally {
145         DBManager.close(con, pstmt, rs);
146     }
147     return oseqList;
148 }
149 }

```

장바구니에 담긴 상품을 주문하기 위해서는 [주문하기] 버튼을 클릭해야 합니다. 그러면 “NonageServlet?command=order\_insert” 요청이 발생합니다. 이 요청을 받아 처리하는 액션을 만들어 봅시다.

**[직접해보세요]** 주문 처리를 위한 액션 클래스

[파일이름 : src\com\Nonage\controller\action\OrderInsertAction.java]

```

1 package com.nonage.controller.action;
2
3 import java.io.IOException;
4 import java.util.ArrayList;
5
6 import javax.servlet.ServletException;
7 import javax.servlet.http.HttpServletRequest;
8 import javax.servlet.http.HttpServletResponse;
9 import javax.servlet.http.HttpSession;
10
11 import com.nonage.dao.CartDAO;
12 import com.nonage.dao.OrderDAO;
13 import com.nonage.dto.CartVO;
14 import com.nonage.dto.MemberVO;
15
16 public class OrderInsertAction implements Action {
17
18     @Override

```

```

19 public void execute(HttpServletRequest request, HttpServletResponse response)
20     throws ServletException, IOException {
21     String url = "NonageServlet?command=order_list";
22
23     HttpSession session = request.getSession();
24     MemberVO loginUser = (MemberVO) session.getAttribute("loginUser");
25     if (loginUser == null) {
26         url = "NonageServlet?command=login_form";
27     } else {
28         CartDAO cartDAO = CartDAO.getInstance();
29         ArrayList<CartVO> cartList = cartDAO.listCart(loginUser.getId());
30
31         OrderDAO orderDAO = OrderDAO.getInstance();
32
33         int maxOseq=orderDAO.insertOrder(cartList, loginUser.getId());
34         url = "NonageServlet?command=order_list&oseq="+maxOseq;
35     }
36     response.sendRedirect(url);
37 }
38 }

```

**[직접해보세요]** 커맨드(command) 패턴으로 작업 처리를 위한 명령 처리 클래스  
ActionFactory 수정하기 [파일 이름 : srcWcomWnonageWcontrollerWActionFactory.java]

```

.. //중복되는 내용 생략
46     } else if (command.equals("order_insert")) {
47         action = new OrderInsertAction();
48     }
.. //중복되는 내용 생략

```

주문 처리하고 나면 "NonageServlet?command=order\_list" 요청이 발생합니다. 이 요청은  
주문한 내역을 출력합니다. 이 요청을 처리할 액션 클래스를 만듭시다.

**[직접해보세요]** 주문 내역 리스트를 처리한 액션 클래스

[파일 이름 : srcWcomWnonageWcontrollerWactionWOrderListAction.java]

```

1 package com.nonage.controller.action;
2
3 import java.io.IOException;
4 import java.util.ArrayList;
5
6 import javax.servlet.ServletException;
7 import javax.servlet.http.HttpServletRequest;
8 import javax.servlet.http.HttpServletResponse;
9 import javax.servlet.http.HttpSession;
10
11 import com.nonage.dao.OrderDAO;

```

```

12 import com.nonage.dto.MemberVO;
13 import com.nonage.dto.OrderVO;
14
15 public class OrderListAction implements Action {
16
17     @Override
18     public void execute(HttpServletRequest request, HttpServletResponse response)
19         throws ServletException, IOException {
20         String url = "mypage/orderList.jsp";
21
22         HttpSession session = request.getSession();
23         MemberVO loginUser = (MemberVO) session.getAttribute("loginUser");
24
25         if (loginUser == null) {
26             url = "NonageServlet?command=login_form";
27         } else {
28             OrderDAO orderDAO = OrderDAO.getInstance();
29             int oseq=Integer.parseInt(request.getParameter("oseq"));
30             ArrayList<OrderVO> orderList =
31 orderDAO.listOrderById(loginUser.getId(), "1", oseq);
32
33             int totalPrice=0;
34             for(OrderVO orderVO :orderList){
35                 totalPrice+=orderVO.getPrice2()*orderVO.getQuantity();
36             }
37
38             request.setAttribute("orderList", orderList);
39             request.setAttribute("totalPrice", totalPrice);
40         }
41         request.getRequestDispatcher(url).forward(request, response);
42     }
43 }

```

**[직접해보세요]** 주문 내역 리스트를 출력하는 JSP 파일

[파일 이름 : WebContent/mypage/orderList.jsp]

```

1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <%@ include file="../header.jsp" %>
4 <%@ include file="sub_img.html"%>
5 <%@ include file="sub_menu.jsp" %>
6 <article>
7   <h2> Order List </h2>
8   <form name="formm" method="post">
9     <table id="cartList">
10      <tr>
11        <th>상품명</th><th>수량</th><th>가격</th><th>주문일</th><th>진행상태</th>
12      </tr>

```

```

13      <c:forEach items="${orderList}" var="orderVO">
14      <tr>
15          <td>
16              <a href="NonageServlet?command=product_detail&pseq=${cartVO.pseq}">
17                  <h3> ${orderVO.pname} </h3>
18              </a>
19          </td>
20          <td> ${orderVO.quantity} </td>
21          <td> <fmt:formatNumber value="${orderVO.price2*orderVO.quantity}"
22              type="currency"/> </td>
23          <td> <fmt:formatDate value="${orderVO.indate}" type="date"/></td>
24          <td> 처리 진행 중 </td>
25      </tr>
26      </c:forEach>
27      <tr>
28          <th colspan="2"> 총 액 </th>
29          <th colspan="2"> <fmt:formatNumber value="${totalPrice}"
30              type="currency"/><br></th>
31          <th> 주문 처리가 완료되었습니다. </th>
32      </tr>
33  </table>
34
35  <div class="clear"></div>
36  <div id="buttons" style="float: right">
37      <input type="button" value="쇼핑 계속하기" class="cancel"
38      onclick="location.href='NonageServlet?command=index'">
39  </div>
40  </form>
41  </article>
42  <%@ include file="../footer.jsp" %>

```

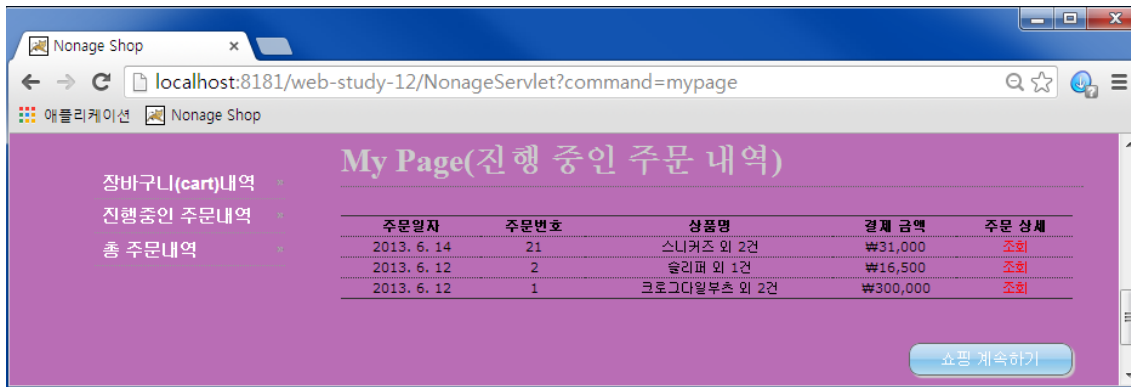
**[직접해보세요]** 커맨드(command) 패턴으로 작업 처리를 위한 명령 처리 클래스  
ActionFactory 수정하기 [파일 이름 : srcWcomWnonageWcontrollerWActionFactory.java]

```

.. //중복되는 내용 생략
48     } else if (command.equals("order_list")) {
49         action = new OrderListAction();
50     }
.. //중복되는 내용 생략

```

현재 주문 진행 중인 목록을 확인하려면 화면 상단의 주 메뉴에서 mypage를 클릭합니다.



mypage를 클릭하면 “NonageServlet?command=mypage” 요청이 발생합니다. 이 요청을 받아 처리하는 액션을 만들어 봅시다.

**[직접해보세요]** 현재 진행 중인 주문 목록을 확인하기 위한 액션 클래스

[파일이름 : srcWcomWnonageWcontrollerWactionWMyPageAction.java]

```

1 package com.nonage.controller.action;
2
3 import java.io.IOException;
4 import java.util.ArrayList;
5
6 import javax.servlet.ServletException;
7 import javax.servlet.http.HttpServletRequest;
8 import javax.servlet.http.HttpServletResponse;
9 import javax.servlet.http.HttpSession;
10
11 import com.nonage.dao.OrderDAO;
12 import com.nonage.dto.MemberVO;
13 import com.nonage.dto.OrderVO;
14
15 public class MyPageAction implements Action {
16
17     @Override
18     public void execute(HttpServletRequest request, HttpServletResponse response)
19         throws ServletException, IOException {
20         String url = "mypage/mypage.jsp";
21
22         HttpSession session = request.getSession();
23         MemberVO loginUser = (MemberVO) session.getAttribute("loginUser");
24
25         if (loginUser == null) {
26             url = "NonageServlet?command=login_form";
27         } else {
28             OrderDAO orderDAO = OrderDAO.getInstance();
29             ArrayList<Integer> oseqList = orderDAO.selectSeqOrderIng(loginUser
30                 .getIdx());

```

```

31
32     ArrayList<OrderVO> orderList = new ArrayList<OrderVO>();
33
34     for (int oseq : oseqList) {
35         ArrayList<OrderVO> orderListIng = orderDAO.listOrderByld(
36             loginUser.getId(), "1", oseq);
37
38         OrderVO orderVO = orderListIng.get(0);
39         orderVO.setPname(orderVO.getPname() + " 외 "
40             + orderListIng.size() + "건");
41
42         int totalPrice = 0;
43         for (OrderVO ovo : orderListIng) {
44             totalPrice += ovo.getPrice2() * ovo.getQuantity();
45         }
46         orderVO.setPrice2(totalPrice);
47         orderList.add(orderVO);
48     }
49     request.setAttribute("title", "진행 중인 주문 내역");
50     request.setAttribute("orderList", orderList);
51 }
52 request.getRequestDispatcher(url).forward(request, response);
53 }
54 }

```

**[직접해보세요]** 현재 진행 중인 주문 목록을 확인하기 위한 JSP 파일

[파일이름 : WebContent/mypage/mypage.jsp]

```

1  <%@ page language="java" contentType="text/html; charset=UTF-8"
2      pageEncoding="UTF-8"%>
3  <%@ include file="../header.jsp" %>
4  <%@ include file="sub_img.html"%>
5  <%@ include file="sub_menu.jsp" %>
6  <article>
7      <h2> My Page(${title}) </h2>
8      <form name="formm" method="post">
9          <table id="cartList">
10             <tr>
11                 <th>주문일자</th> <th>주문번호</th> <th>상품명</th>
12 <th>결제 금액</th> <th>주문 상세</th> </th>
13             </tr>
14             <c:forEach items="${orderList}" var="orderVO">
15                 <tr>
16                     <td> <fmt:formatDate value="${orderVO.indate}" type="date"/></td>
17                     <td> ${orderVO.oseq} </td>
18                     <td> ${orderVO.pname} </td>
19                     <td> <fmt:formatNumber value="${orderVO.price2}" type="currency"/> </td>
20                     <td> <a href=

```



```

21 "NonageServlet?command=order_detail&oseq=${orderVO.oseq}"> 조회 </a></td>
22     </tr>
23 </c:forEach>
24 </table>
25
26 <div class="clear"></div>
27 <div id="buttons" style="float: right">
28     <input type="button" value="쇼핑 계속하기" class="cancel"
29 onclick="location.href='NonageServlet?command=index'">
30 </div>
31 </form>
32 </article>
33 <%@ include file="../footer.jsp" %>

```

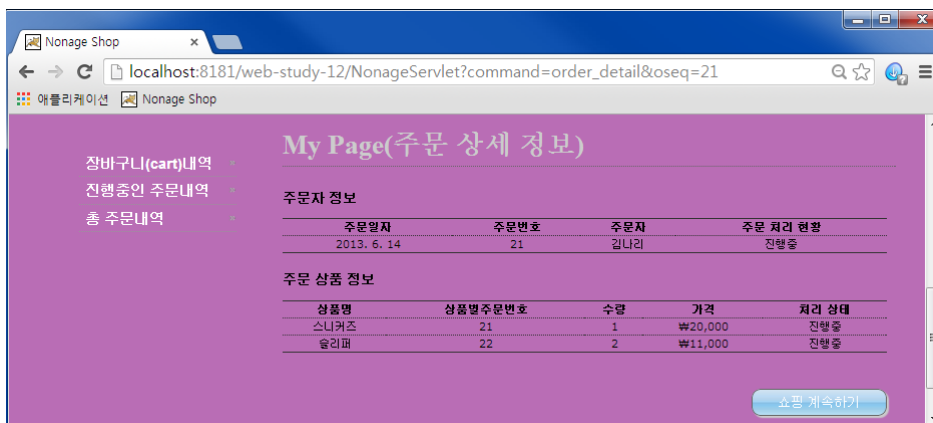
**[직접해보세요]** 커맨드(command) 패턴으로 작업 처리를 위한 명령 처리 클래스  
ActionFactory 수정하기[파일이름 : srcWcomWnonageWcontrollerWActionFactory.java]

```

.. //중복되는 내용 생략
50     } else if (command.equals("mypage")) {
51         action = new MyPageAction();
52     }
.. //중복되는 내용 생략

```

여러 건에 대한 주문을 한 번에 하기 때문에 주문에 대한 상세보기를 위한 페이지를 작성해  
서 주문한 날짜와 물품에 대해서 확인할 수 있도록 합니다.



주문 목록을 보다가 주문 상세 정보를 보고 싶다면 화면 왼쪽의 조회를 클릭합니다. 그러면  
“ NonageServlet?command=order\_detail” 요청이 발생합니다. 이 요청을 처리할 액션 클래스  
를 만듭니다.

**[직접해보세요]** 주문 상세 정보를 보기 위한 액션 클래스  
[파일이름 : srcWcomWnonageWcontrollerWactionWOrderDetailAction.java]

```

1 package com.nonage.controller.action;
2

```

```

3 import java.io.IOException;
4 import java.util.ArrayList;
5
6 import javax.servlet.ServletException;
7 import javax.servlet.http.HttpServletRequest;
8 import javax.servlet.http.HttpServletResponse;
9 import javax.servlet.http.HttpSession;
10
11 import com.nonage.dao.OrderDAO;
12 import com.nonage.dto.MemberVO;
13 import com.nonage.dto.OrderVO;
14
15 public class OrderDetailAction implements Action {
16
17     @Override
18     public void execute(HttpServletRequest request, HttpServletResponse response)
19         throws ServletException, IOException {
20         String url = "mypage/orderDetail.jsp";
21
22         HttpSession session = request.getSession();
23         MemberVO loginUser = (MemberVO) session.getAttribute("loginUser");
24         if (loginUser == null) {
25             url = "NonageServlet?command=login_form";
26         } else {
27             int oseq=Integer.parseInt(request.getParameter("oseq"));
28             OrderDAO orderDAO = OrderDAO.getInstance();
29             ArrayList<OrderVO> orderList =
30 orderDAO.listOrderByld(loginUser.getId(), "%", oseq);
31
32             int totalPrice=0;
33             for(OrderVO ovo :orderList){
34                 totalPrice+=ovo.getPrice2()*ovo.getQuantity();
35             }
36             request.setAttribute("orderDetail", orderList.get(0));
37             request.setAttribute("orderList", orderList);
38             request.setAttribute("totalPrice", totalPrice);
39         }
40         request.getRequestDispatcher(url).forward(request, response);
41     }
42 }

```

**[직접해보세요]** 주문 상세 보기 페이지를 출력하는 JSP 파일

[파일이름 : WebContent/mypage/orderDetail.jsp]

```

1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <%@ include file="../header.jsp" %>
4 <%@ include file="sub_img.html"%>

```

```

5 <%@ include file="sub_menu.jsp" %>
6 <article>
7 <h2> My Page(주문 상세 정보) </h2>
8 <form name="formm" method="post">
9 <h3> 주문자 정보 </h3>
10 <table id="cartList">
11 <tr>
12 <th>주문일자</th> <th>주문번호</th> <th>주문자</th>
13 <th> 주문 총액</th>
14 </tr>
15 <tr>
16 <td> <fmt:formatDate value="${orderDetail.indate}" type="date"/></td>
17 <td> ${orderDetail.oseq} </td>
18 <td> ${orderDetail.mname} </td>
19 <td> <fmt:formatNumber type="currency"
20 value="${totalPrice}" /> </td>
21 </tr>
22 </table>
23 <h3> 주문 상품 정보 </h3>
24 <table id="cartList">
25 <tr>
26 <th>상품명</th> <th>상품별주문번호</th> <th>수량</th>
27 <th>가격</th> <th>처리 상태</th>
28 </tr>
29 <c:forEach items="${orderList}" var="orderVO">
30 <tr>
31 <td> ${orderVO.pname} </td>
32 <td> ${orderVO.odseq} </td>
33 <td> ${orderVO.quantity} </td>
34 <td> <fmt:formatNumber type="currency"
35 value="${orderVO.price2*orderVO.quantity}" /> </td>
36 <td>
37 <c:choose>
38 <c:when test='${orderVO.result=="1"}> 진행중 </c:when>
39 <c:otherwise> <span style="color:red"> 처리완료 </span></c:otherwise>
40 </c:choose>
41 </td>
42 </tr>
43 </c:forEach>
44 </table>
45 <div class="clear"></div>
46 <div id="buttons" style="float: right">
47 <input type="button" value="쇼핑 계속하기" class="cancel"
48 onclick="location.href='NonageServlet?command=index'">
49 </div>
50 </form>
51 </article>
52 <%@ include file="../footer.jsp" %>

```

**[직접해보세요]** 커맨드(command) 패턴으로 작업 처리를 위한 명령 처리 클래스  
ActionFactory 수정하기[파일이름 : srcWcomWnonageWcontrollerWActionFactory.java]

```
.. //중복되는 내용 생략
52     } else if (command.equals("order_detail")) {
53         action = new OrderDetailAction();
54     }
.. //중복되는 내용 생략
```

지금까지 주문한 모든 주문 정보를 보려면 총 주문내역 메뉴를 클릭합니다.  
13-005

그러면 “NonageServlet?command=order\_all” 요청이 발생합니다. 이 요청을 처리할 액션  
클래스를 만듭시다.

**[직접해보세요]** 주문 상세 정보를 보기 위한 액션 클래스  
[파일이름 : srcWcomWnonageWcontrollerWactionWOrderDetailction.java]

```
1 package com.nonage.controller.action;
2
3 import java.io.IOException;
4 import java.util.ArrayList;
5
6 import javax.servlet.ServletException;
7 import javax.servlet.http.HttpServletRequest;
8 import javax.servlet.http.HttpServletResponse;
9 import javax.servlet.http.HttpSession;
10
11 import com.nonage.dao.OrderDAO;
12 import com.nonage.dto.MemberVO;
13 import com.nonage.dto.OrderVO;
14
15 public class OrderAllAction implements Action {
16
17     @Override
18     public void execute(HttpServletRequest request, HttpServletResponse response)
19         throws ServletException, IOException {
20         String url = "mypage/mypage.jsp";
21
22         HttpSession session = request.getSession();
23         MemberVO loginUser = (MemberVO) session.getAttribute("loginUser");
24         if (loginUser == null) {
25             url = "NonageServlet?command=login_form";
26         } else {
27             OrderDAO orderDAO = OrderDAO.getInstance();
28             ArrayList<Integer> oseqList =
```

```

29         orderDAO.selectSeqOrderIng(loginUser.getId());
30
31         ArrayList<OrderVO> orderList = new ArrayList<OrderVO>();
32
33         for (int oseq : oseqList) {
34             ArrayList<OrderVO> orderListIng = orderDAO.listOrderById(
35                 loginUser.getId(), "%", oseq);
36
37             OrderVO orderVO = orderListIng.get(0);
38             orderVO.setPname(orderVO.getPname() + " 외 "
39                 + orderListIng.size() + "건");
40             System.out.println(orderVO.getIndate());
41
42             int totalPrice = 0;
43             for (OrderVO ovo : orderListIng) {
44                 totalPrice += ovo.getPrice2() * ovo.getQuantity();
45             }
46             orderVO.setPrice2(totalPrice);
47             orderList.add(orderVO);
48         }
49         request.setAttribute("title", "총 주문 내역");
50         request.setAttribute("orderList", orderList);
51     }
52     request.getRequestDispatcher(url).forward(request, response);
53 }
54 }

```

**[직접해보세요]** 커맨드(command) 패턴으로 작업 처리를 위한 명령 처리 클래스  
ActionFactory 수정하기 [파일이름 : src\com\wonage\controller\ActionFactory.java]

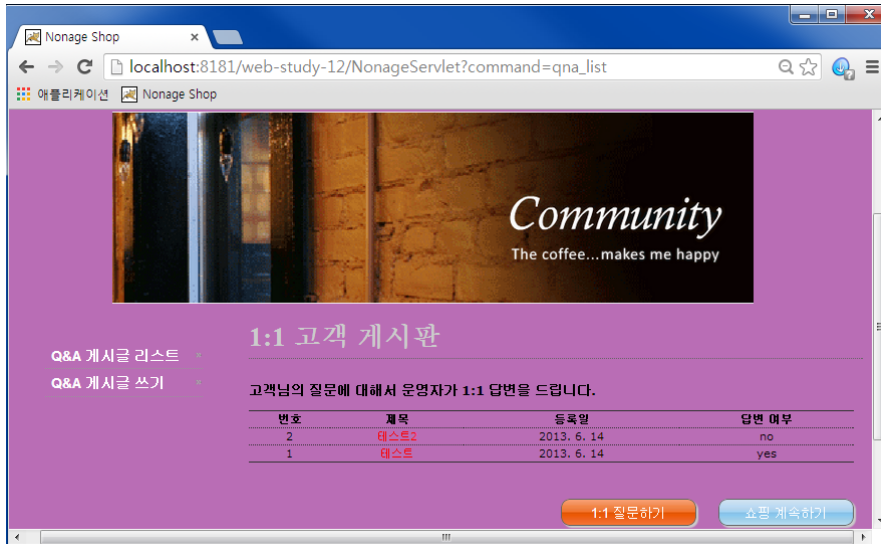
```

.. //중복되는 내용 생략
54     } else if (command.equals("order_all")) {
55         action = new OrderAllAction();
56     }
.. //중복되는 내용 생략

```

## 13-4 Q&A 게시판

이번 장에서 마지막으로 고객이 제품 또는 배송, 반품 등에 대한 전반적인 문의 사항을 기재하고 그에 해당하는 답변을 듣는 Q&A 게시판을 작성해 봅시다.



고객과 관리자 사이에 1:1 게시판이기 때문에 자신이 올린 내역만 출력합니다. Q&A 게시글을 쓰기 위해서는 데이터베이스의 qna 테이블에 정보를 저장해야 합니다. 그러기 위해서는 qna 테이블을 처리하기 위한 DAO를 작성해야 합니다. Q&A 게시판을 위한 DAO를 만들기 위해서 게시판 정보를 저장할 VO부터 만듭시다.

[직접해보세요] 이클립스에서 게시글 정보를 저장하는 VO 클래스 만들기

[파일 이름 : comWnonageWdtoWQnaVO.java]

```
1 package com.nonage.dto;
2
3
4 import java.sql.Timestamp;
5
6 public class QnaVO {
7     private int    qseq;
8     private String subject;
9     private String content;
10    private String reply;
11    private String id;
12    private String rep;
13    private Timestamp indate;
14
15    public int getQseq() {
```

```

19     return qseq;
20 }
21
22 public void setQseq(int qseq) {
23     this.qseq = qseq;
24 }
25
26 public String getReply() {
27     return reply;
28 }
29
30 public void setReply(String reply) {
31     this.reply = reply;
32 }
33
34 public String getSubject() {
35     return subject;
36 }
37
38 public void setSubject(String subject) {
39     this.subject = subject;
40 }
41
42 public String getContent() {
43     return content;
44 }
45
46 public void setContent(String content) {
47     this.content = content;
48 }
49
50 public String getId() {
51     return id;
52 }
53
54 public void setId(String id) {
55     this.id = id;
56 }
57
58 public String getRep() {
59     return rep;
60 }
61
62 public void setRep(String rep) {
63     this.rep = rep;
64 }
65
66 public Timestamp getIndate() {
67     return indate;
68 }

```

```

67     public void setIndate(Timestamp indate) {
68         this.indate = indate;
69     }
70 }

```

**[직접해보세요]** 게시글 테이블을 액세스하는 DAO 클래스 만들기

[파일 이름 :src\com\wonage\dao\QnaDAO.java]

```

1  package com.nonage.dao;
2
3  import java.sql.Connection;
4  import java.sql.PreparedStatement;
5  import java.sql.ResultSet;
6  import java.util.ArrayList;
7
8  import util.DBManager;
9  import com.nonage.dto.QnaVO;
10
11 public class QnaDAO {
12
13     private QnaDAO() {
14     }
15
16     private static QnaDAO instance = new QnaDAO();
17
18     public static QnaDAO getInstance() {
19         return instance;
20     }
21
22     public ArrayList<QnaVO> listQna(String id) {
23         ArrayList<QnaVO> qnaList = new ArrayList<QnaVO>();
24         String sql = "select * from qna where id=? order by qseq desc";
25
26         Connection conn = null;
27         PreparedStatement pstmt = null;
28         ResultSet rs = null;
29         try {
30             conn = DBManager.getConnection();
31             pstmt = conn.prepareStatement(sql);
32             pstmt.setString(1, id);
33             rs = pstmt.executeQuery();
34             while (rs.next()) {
35                 QnaVO qnaVO = new QnaVO();
36                 qnaVO.setQseq(rs.getInt("qseq"));
37                 qnaVO.setSubject(rs.getString("subject"));
38                 qnaVO.setContent(rs.getString("content"));
39                 qnaVO.setIdx(rs.getString("id"));

```



```

40     qnaVO.setIndate(rs.getTimestamp("indate"));
41     qnaVO.setReply(rs.getString("reply"));
42     qnaVO.setRep(rs.getString("rep"));
43     qnaList.add(qnaVO);
44 }
45 } catch (Exception e) {
46     e.printStackTrace();
47 }
48 return qnaList;
49 }
50
51 public QnaVO getQna(int seq) {
52     QnaVO qnaVO = null;
53     String sql = "select * from qna where qseq=?";
54     Connection conn = null;
55     PreparedStatement pstmt = null;
56     ResultSet rs = null;
57     try {
58         conn = DBManager.getConnection();
59         pstmt = conn.prepareStatement(sql);
60         pstmt.setInt(1, seq);
61         rs = pstmt.executeQuery();
62         if (rs.next()) {
63             qnaVO = new QnaVO();
64             qnaVO.setQseq(seq);
65             qnaVO.setSubject(rs.getString("subject"));
66             qnaVO.setContent(rs.getString("content"));
67             qnaVO.setId(rs.getString("id"));
68             qnaVO.setIndate(rs.getTimestamp("indate"));
69             qnaVO.setReply(rs.getString("reply"));
70             qnaVO.setRep(rs.getString("rep"));
71         }
72     } catch (Exception e) {
73         e.printStackTrace();
74     } finally {
75         DBManager.close(conn, pstmt);
76     }
77     return qnaVO;
78 }
79
80 public void insertqna(QnaVO qnaVO, String session_id){
81     String sql = "insert into qna (qseq, subject, " +
82         "content, id) values(qna_seq.nextval, ?, ?, ?)";
83
84     Connection conn = null;
85     PreparedStatement pstmt = null;
86     try {
87         conn = DBManager.getConnection();

```

```

88      pstmt = conn.prepareStatement(sql);
89      pstmt.setString(1, qnaVO.getSubject());
90      pstmt.setString(2, qnaVO.getContent());
91      pstmt.setString(3, session_id);
92      pstmt.executeUpdate();
93  } catch (Exception e) {
94      e.printStackTrace();
95  } finally {
96      DBManager.close(conn, pstmt);
97  }
98  }

```

게시글 리스트를 확인하기 위해서는 “NonageServlet?command=qna\_list”와 같은 요청이 발생합니다. 이 요청을 받으면 게시글 리스트 화면을 표시하기 위한 액션 클래스를 만듭니다.

**[직접해보세요]** 게시글 리스트를 위한 액션 클래스

[파일이름 : srcWcomWnonageWcontrollerWactionWQnaListActoin.java]

```

1  package com.nonage.controller.action;
2
3  import java.io.IOException;
4  import java.util.ArrayList;
5
6  import javax.servlet.ServletException;
7  import javax.servlet.http.HttpServletRequest;
8  import javax.servlet.http.HttpServletResponse;
9  import javax.servlet.http.HttpSession;
10
11 import com.nonage.dao.QnaDAO;
12 import com.nonage.dto.MemberVO;
13 import com.nonage.dto.QnaVO;
14
15 public class QnaListAction implements Action {
16
17     @Override
18     public void execute(HttpServletRequest request, HttpServletResponse response)
19         throws ServletException, IOException {
20         String url = "qna/qnaList.jsp";
21
22         HttpSession session = request.getSession();
23         MemberVO loginUser = (MemberVO) session.getAttribute("loginUser");
24
25         if (loginUser == null) {
26             url = "NonageServlet?command=login_form";
27         } else {
28             QnaDAO qnaDAO = QnaDAO.getInstance();

```

```

29     ArrayList<QnaVO> qnaList = qnaDAO.listQna(loginUser.getId());
30     request.setAttribute("qnaList", qnaList);
31 }
32 request.getRequestDispatcher(url).forward(request, response);
33 }
34 }

```

게시글 리스트 위한 페이지는 qna 폴더에 추가해야 하므로 합니다. qna 폴더를 만든 후에 qnaList.jsp 페이지를 이 폴더에 추가합니다.

**[직접해보세요]** 게시글 리스트를 위한 JSP 페이지

[파일이름 : WebContent/qna/qnaList.jsp]

```

1  <%@ page language="java" contentType="text/html; charset=UTF-8"
2      pageEncoding="UTF-8"%>
3  <%@ include file="../header.jsp" %>
4  <%@ include file="sub_img.html"%>
5  <%@ include file="sub_menu.jsp" %>
6  <article>
7      <h2> 1:1 고객 게시판 </h2>
8      <h3> 고객님의 질문에 대해서 운영자가 1:1 답변을 드립니다.</h3>
9      <form name="formm" method="post">
10         <table id="cartList">
11             <tr>
12                 <th>번호</th> <th>제목</th> <th>등록일</th> <th>답변 여부</th>
13             </tr>
14             <c:forEach items="${qnaList}" var="qnaVO">
15                 <tr>
16                     <td> ${qnaVO.qseq} </td>
17                     <td> <a href="NonageServlet?command=qna_view&qseq=${qnaVO.qseq}">
18 ${qnaVO.subject} </td>
19                     <td> <fmt:formatDate value="${qnaVO.indate}" type="date"/></td>
20                     <td>
21                         <c:choose>
22                             <c:when test="${qnaVO.rep==1}"> no </c:when>
23                             <c:when test="${qnaVO.rep==2}"> yes </c:when>
24                         </c:choose>
25                     </td>
26                 </tr>
27             </c:forEach>
28         </table>
29         <div class="clear"></div>
30         <div id="buttons" style="float:right">
31             <input type="button" value="1:1 질문하기" class="submit"
32 onclick="location.href='NonageServlet?command=qna_write_form'">
33             <input type="button" value="쇼핑 계속하기" class="cancel"
34 onclick="location.href='NonageServlet?command=index'">

```

```

35     </div>
36     </form>
37     </article>
38     <%@ include file="../footer.jsp" %>

```

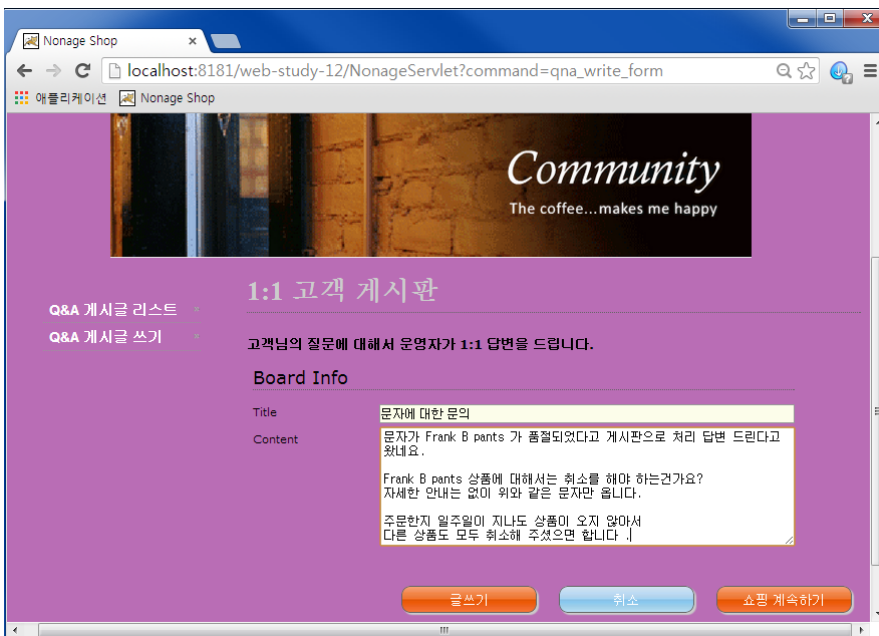
**[직접해보세요]** 커맨드(command) 패턴으로 작업 처리를 위한 명령 처리 클래스  
 ActionFactory 수정하기[파일이름 : srcWcomWnonageWcontroller WActionFactory.java]

```

.. //중복되는 내용 생략
56     } else if(command.equals("qna_list")){
57         action = new QnaListAction();
58     }
.. //중복되는 내용 생략

```

[1:1 질문하기] 버튼을 누르면 게시물 등록 화면으로 이동합니다.



[1:1 질문하기] 버튼을 클릭하면 "NonageServlet?command=qna\_write\_form"이 요청되어  
 게시물 등록 페이지로 이동해야 하기 때문에 액션 클래스를 만들고 이 요청을 처리할 코드  
 를 ActionFactory에 추가해야 합니다. 액션 클래스에서는 게시물 등록 화면으로 이동하도록  
 합니다.

**[직접해보세요]** 게시물 등록을 위한 폼으로 이동하도록 하는 액션 클래스[파일이름 :  
 srcWcomWnonageWcontrollerWactionWQnaWriteFormAction.java]

```

1 package com.nonage.controller.action;
2
3 import java.io.IOException;
4
5 import javax.servlet.ServletException;

```

```

6 import javax.servlet.http.HttpServletRequest;
7 import javax.servlet.http.HttpServletResponse;
8 import javax.servlet.http.HttpSession;
9
10 import com.nonage.dto.MemberVO;
11
12
13 public class QnaWriteFormAction implements Action {
14
15     @Override
16     public void execute(HttpServletRequest request, HttpServletResponse response)
17         throws ServletException, IOException {
18         String url = "qna/qnaWrite.jsp";
19
20         HttpSession session = request.getSession();
21         MemberVO loginUser = (MemberVO) session.getAttribute("loginUser");
22
23         if (loginUser == null) {
24             url = "NonageServlet?command=login_form";
25         }
26
27         request.getRequestDispatcher(url).forward(request, response);
28     }
29 }

```

**[직접해보세요]** 게시글 등록 화면을 위한 JSP 페이지

[파일이름 : WebContent/qna/qnaWrite.jsp]

```

1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <%@ include file="../header.jsp" %>
4 <%@ include file="sub_img.html"%>
5 <%@ include file="sub_menu.jsp" %>
6 <article>
7   <h2> 1:1 고객 게시판 </h2>
8   <h3> 고객님의 질문에 대해서 운영자가 1:1 답변을 드립니다.</h3>
9   <form name="formm" method="post"
10   action="NonageServlet?command=qna_write">
11     <fieldset>
12       <legend>Qna Info</legend>
13       <label>Title</label>
14       <input type="text" name="subject" size="77" ><br>
15       <label>Content</label>
16       <textarea rows="8" cols="65" name="content"></textarea><br>
17     </fieldset>
18     <div class="clear"></div>
19     <div id="buttons" style="float:right">
20       <input type="submit" value="글쓰기" class="submit">

```

```

21      <input type="reset" value="취소" class="cancel">
22      <input type="button" value="쇼핑 계속하기" class="submit"
23      onclick="location.href='NonageServlet?command=index'">
24      </div>
25      </form>
26      </article>
27      <%@ include file="../footer.jsp" %>

```

**[직접해보세요]** 커맨드(command) 패턴으로 작업 처리를 위한 명령 처리 클래스  
ActionFactory 수정하기 [파일 이름 : srcWcomWnonageWcontrollerWActionFactory.java]

```

..      //중복되는 내용 생략
58      } else if (command.equals("qna_write_form")) {
59          action = new QnaWriteFormAction();
60      }
..      //중복되는 내용 생략

```

게시글 정보를 입력하고 [글쓰기] 버튼을 클릭하면 "NonageServlet?command=qna\_write"  
요청이 발생합니다. 이 요청을 처리할 코드를 액션 클래스를 만듭시다.

**[직접해보세요]** 게시글을 데이터베이스에 추가하기 위한 액션 클래스  
[파일 이름 : srcWcomWnonageWcontroller WactionWQnaWriteAction.java]

```

1  package com.nonage.controller.action;
2
3  import java.io.IOException;
4
5  import javax.servlet.ServletException;
6  import javax.servlet.http.HttpServletRequest;
7  import javax.servlet.http.HttpServletResponse;
8  import javax.servlet.http.HttpSession;
9
10 import com.nonage.dao.QnaDAO;
11 import com.nonage.dto.MemberVO;
12 import com.nonage.dto.QnaVO;
13
14 public class QnaWriteAction implements Action {
15
16     @Override
17     public void execute(HttpServletRequest request, HttpServletResponse response)
18         throws ServletException, IOException {
19         String url = "NonageServlet?command=qna_list";
20
21         HttpSession session = request.getSession();
22         MemberVO loginUser = (MemberVO) session.getAttribute("loginUser");
23
24         if (loginUser == null) {

```

```

25     url = "NonageServlet?command=login_form";
26 }else{
27     QnaVO qnaVO = new QnaVO();
28     qnaVO.setSubject(request.getParameter("subject"));
29     qnaVO.setContent(request.getParameter("content"));
30     QnaDAO qnaDAO = QnaDAO.getInstance();
31     qnaDAO.insertqna(qnaVO, loginUser.getId());
32 }
33 response.sendRedirect(url);
34 }
35 }

```

**[직접해보세요]** 커맨드(command) 패턴으로 작업 처리를 위한 명령 처리 클래스  
 ActionFactory 수정하기 [파일 이름 : srcWcomWnonageWcontroller WActionFactory.java]

```

.. //중복되는 내용 생략
60     } else if (command.equals("qna_write")) {
61         action = new QnaWriteAction();
62     }
... //중복되는 내용 생략

```

게시글 제목을 클릭하면 게시글 상세보기 화면으로 이동합니다. 게시글 제목을 클릭하면  
 "NonageServlet?command=qna\_view"가 요청되어 게시글 상세 보기 페이지로 이동해야 하  
 기 때문에 이 요청을 처리할 코드를 액션 클래스를 만듭시다.

**[직접해보세요]** 게시글 상세 보기 페이지로 이동하도록 하는 액션 클래스  
 [파일 이름 : srcWcomWnonageWcontroller WactionWQnaViewAction.java]

```

1  package com.nonage.controller.action;
2
3  import java.io.IOException;
4
5  import javax.servlet.ServletException;
6  import javax.servlet.http.HttpServletRequest;
7  import javax.servlet.http.HttpServletResponse;
8  import javax.servlet.http.HttpSession;
9
10 import com.nonage.dao.QnaDAO;
11 import com.nonage.dto.MemberVO;
12 import com.nonage.dto.QnaVO;
13
14 public class QnaViewAction implements Action {
15
16     @Override
17     public void execute(HttpServletRequest request, HttpServletResponse response)
18         throws ServletException, IOException {
19         String url = "qna/qnaView.jsp";
20

```

```

21 HttpSession session = request.getSession();
22 MemberVO loginUser = (MemberVO) session.getAttribute("loginUser");
23
24 if (loginUser == null) {
25     url = "NonageServlet?command=login_form";
26 } else {
27     int qseq = Integer.parseInt(request.getParameter("qseq"));
28     QnaDAO qnaDAO = QnaDAO.getInstance();
29     QnaVO qnaVO = qnaDAO.getQna(qseq);
30     request.setAttribute("qnaVO", qnaVO);
31 }
32 request.getRequestDispatcher(url).forward(request, response);
33 }
34 }

```

**[직접해보세요]** 게시글 상세 보기를 위한 JSP 페이지

[파일이름 : WebContent/qna/qnaView.jsp]

```

1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <%@ include file="../header.jsp" %>
4 <%@ include file="sub_img.html"%>
5 <%@ include file="sub_menu.jsp" %>
6 <article>
7   <h2> 1:1 고객 게시판 </h2>
8   <h3> 고객님의 질문에 대해서 운영자가 1:1 답변을 드립니다.</h3>
9   <form name="formm" method="post">
10    <table id="notice">
11      <tr>
12        <th>제목</th>
13        <td>${qnaVO.subject}</td>
14      </tr>
15      <tr>
16        <th>등록일</th>
17        <td> <fmt:formatDate value="${qnaVO.indate}" type="date"/></td>
18      </tr>
19      <tr>
20        <th>질문내용</th>
21        <td>${qnaVO.content}</td>
22      </tr>
23      <tr>
24        <th>답변 내용</th>
25        <td>${qnaVO.reply}</td>
26      </tr>
27    </table>
28    <div class="clear"></div>
29    <div id="buttons" style="float:right">
30      <input type="button" value="목록보기" class="submit"

```



```

31 onclick="location.href='NonageServlet?command=qna_list'">
32     <input type="button" value="쇼핑 계속하기" class="cancel"
33 onclick="location.href='NonageServlet?command=index'">
34     </div>
35 </form>
36 </article>
37 <%@ include file="../footer.jsp" %>

```

**[직접해보세요]** 커맨드(command) 패턴으로 작업 처리를 위한 명령 처리 클래스  
 ActionFactory 수정하기 [파일 이름 : srcWcomWnonageWcontroller WActionFactory.java]

```

.. //중복되는 내용 생략
25     } else if (command.equals("qna_view")) {
26         action = new QnaViewAction();
27     }
28
.. //중복되는 내용 생략

```