

14장.

쇼핑몰-관리자 페이지

관리자 페이지에서는 상품 관리, 주문 관리, 회원 관리, 고객이 올린 질문에 대한 답변을 다는 등 다양한 관리를 하게 됩니다. 이번 장에서는 쇼핑몰 관리를 위한 관리자 페이지를 작성해 보도록 합시다.

14-1 전반적인 흐름 분석하기

14-2 상품 관리

14-3 주문 관리

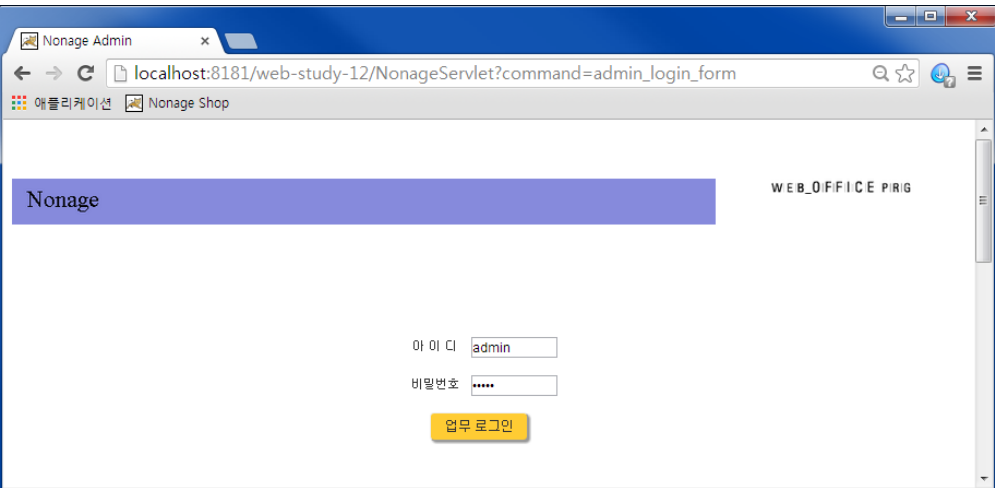
14-4 회원 관리

14-5 Q&A 게시판 관리

14-1 전반적인 흐름 분석하기

관리자 페이지로 들어가기 위해서는 가장 먼저 관리자 인증 처리를 해야 합니다. 그래서 관리자 페이지의 첫 시작은 관리자인지를 확인하기 위해 아이디와 비밀번호를 입력받도록 하는 로그인 페이지부터 시작합니다.

▼ 관리자 로그인 인증 화면



관리자인 경우 로그인에 성공하고 상품 관리, 주문 관리, 회원 관리, Q&A 게시판 관리 등의 기능을 제공합니다.

▼ 상품 관리



▼ 주문 관리

Nonage Admin

localhost:8181/web-study-12/NonageServlet?command=admin_order_list

Nonage

WEB_OFFICE PRG

logout

Admin Setting

주문리스트

상품리스트 주문자 이름 검색

주문번호(처리여부)	주문지	상품명	수량	우편번호	배송지	전화	주문일
4 (미처리)	이백합	밀	1	130-120	서울시송파구잠실2동 리렌츠 아파트 201동 505호	011-123-4567	2013. 6. 15
7 (미처리)	이백합	크로그디일부츠	2	130-120	서울시송파구잠실2동 리렌츠 아파트 201동 505호	011-123-4567	2013. 6. 15
3 (미처리)	김나리	올리퍼	3	133-110	서울시성동구성수동1가 1번지21호	017-777-7777	2013. 6. 15
1 (미처리)	김나리	크로그디일부츠	1	133-110	서울시성동구성수동1가 1번지21호	017-777-7777	2013. 6. 15
2 (미처리)	김나리	롱부츠	5	133-110	서울시성동구성수동1가 1번지21호	017-777-7777	2013. 6. 15
5 (처리완료)	이백합	롱부츠	1	130-120	서울시송파구잠실2동 리렌츠 아파트 201동 505호	011-123-4567	2013. 6. 15
6 (처리완료)	이백합	여성부츠	2	130-120	서울시송파구잠실2동 리렌츠 아파트 201동 505호	011-123-4567	2013. 6. 15

주문처리(입금확인)

▼ 회원 관리

Nonage Admin

localhost:8181/web-study-12/NonageServlet?command=admin_member_list

Nonage

WEB_OFFICE PRG

logout

Admin Setting

회원리스트

회원 이름 검색

메디디(탈퇴여부)	이름	이메일	우편번호	주소	전화	가입일
two	이백합		130-120	서울시송파구잠실2동 리렌츠 아파트 201동 505호	011-123-4567	2013. 6. 15
one	김나리		133-110	서울시성동구성수동1가 1번지21호	017-777-7777	2013. 6. 15

▼ Q&A 게시판 관리

Nonage Admin

localhost:8181/web-study-12/NonageServlet?command=admin_qna_list#

Nonage

WEB_OFFICE PRG

logout

Admin Setting

Q&A 게시판 리스트

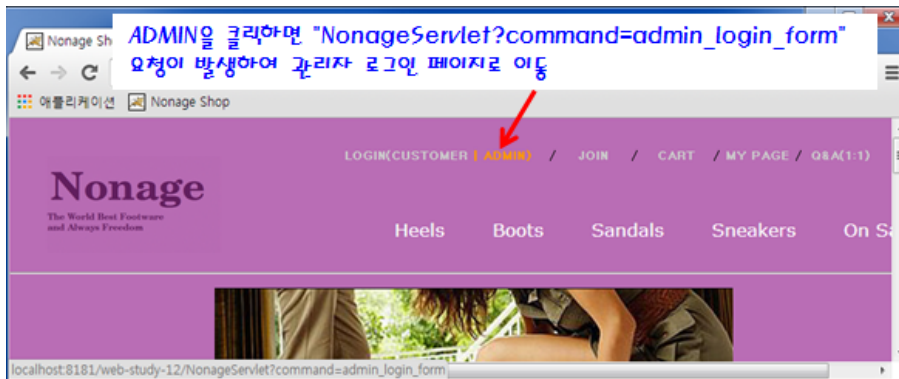
번호(답변여부)	제목	작성자	작성일
2 (미처리)	테스트2	one	2013. 6. 17
1 (답변처리완료)	테스트	one	2013. 6. 17

관리자로 로그인하기 위한 관리자 로그인 페이지로 이동하기 위해서 이전에 작성했던 header.jsp 페이지의 일부를 수정해서 관리자 페이지로 이동하는 링크를 추가합니다.

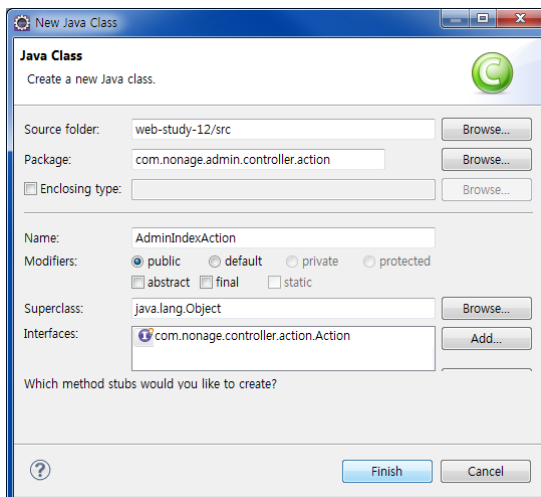
```

29 <c:when test="${empty sessionScope.loginUser}">
30 <li>
31 <a href="/NonageServlet?command=login_form" style="width:110px;">LOGIN(CUSTOMER</a>
32 <a href="/NonageServlet?command=admin_login_form" style="width:100px;">| ADMIN</a>
33 </li>
34 </li></li>
35 <li><a href="/NonageServlet?command=contract">JOIN</a></li>
36 </c:when>

```



관리자 로그인 페이지로 이동하는 “ADMIN”을 클릭하면 "NonageServlet?command=admin_login_form" 요청이 발생합니다. 이 요청을 받아 처리하는 액션을 만들어 봅시다. 관리자 페이지에서 발생하는 요청을 처리하는 Action 클래스는 사용자와는 별도로 “com.nonage.admin.controller.action” 패키지에 만듭시다.



[직접해보세요] 관리자 로그인 페이지로 이동하기 위한 액션 클래스[파일이름 : srcWcomWnonageWadminWcontrollerWactionWAdminIndexAction.java]

```

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

```

```

6 import javax.servlet.ServletException;
7 import javax.servlet.http.HttpServletRequest;
8 import javax.servlet.http.HttpServletResponse;
9
10 import com.nonage.controller.action.Action;
11
12 public class AdminIndexAction implements Action {
13
14     @Override
15     public void execute(HttpServletRequest request, HttpServletResponse response)
16         throws ServletException, IOException {
17         String url = "admin/main.jsp";
18
19         RequestDispatcher dispatcher=request.getRequestDispatcher(url);
20         dispatcher.forward(request, response);
21     }
22 }

```

관리자를 위한 페이지는 사용자 페이지와는 별도로 admin 폴더에 추가하도록 합니다.
admin 폴더를 만든 후에 main.jsp 페이지를 이 폴더에 추가합니다.

[직접해보세요] 관리자 로그인을 위한 JSP 페이지[파일 이름 : WebContent/admin/main.jsp]

```

1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
4 <%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
5 <!DOCTYPE html>
6 <html>
7 <head>
8 <meta charset="UTF-8">
9 <title>Nonage Admin</title>
10 <link rel="stylesheet" href="admin/css/admin.css">
11 <script type="text/javascript">
12 function worker_check()
13 {
14     if(document.frm.workId.value==""){
15         alert("아이디를 입력하세요.");
16         return false;
17     }else if(document.frm.workPw.value==""){
18         alert("비밀번호를 입력하세요.");
19         return false;
20     }
21     return true;
22 }
23 </script>

```

```

24 </head>
25
26 <body>
27     <div id="wrap">
28         <header>
29             <div id="logo">
30                 <a href="admin/main.jsp">
31                     
32                     
33                 </a>
34             </div>
35         </header>
36         <div class="clear"></div>
37         <article>
38             <div id="loginform">
39                 <form name="frm" method="post"
40 action="NonageServlet?command=admin_login">
41                     <table>
42                         <tr>
43                             <td> 아이디 </td>
44                             <td> <input type="text" name="workerId" size="10" value="admin"></td>
45                         </tr>
46                         <tr>
47                             <td> 비밀번호 </td>
48                             <td>
49                                 <input type="password" name="workerPwd" size="10" value="admin">
50                             </td>
51                         </tr>
52                         <tr align="center" >
53                             <td colspan="2">
54                                 <input class="btn" type="submit" value="업무 로그인"
55 onclick="return worker_check()"><br><br>
56                                 <h4 style="color:red">${message}</h4>
57                             </td>
58                         </tr>
59                     </table>
60                 </form>
61             </div>
62         </article>
63     </div>
64 </body>
65 </html>

```

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

```

.. //중복되는 내용 생략
66 //admin

```

```

67     if (command.equals("admin_login_form")) {
68         action = new AdminIndexAction();
69     }
..    //중복되는 내용 생략

```

로그인 처리를 위해서 관리자 정보를 담고 있는 worker 테이블을 조회해야 하기 때문에 DAO 클래스를 만들어야 합니다. 관리자 인증을 위한 DAO를 만듭시다.

[직접해보세요] 관리자 인증을 위한 DAO 클래스 만들기 [파일 이름 :srcWcomWnonageWdaoWWorkerDAO.java]

```

1  package com.nonage.dao;
2
3  import java.sql.Connection;
4  import java.sql.PreparedStatement;
5  import java.sql.ResultSet;
6  import util.DBManager;
7
8  public class WorkerDAO {
9      private WorkerDAO() {
10     }
11
12     private static WorkerDAO instance = new WorkerDAO();
13
14     public static WorkerDAO getInstance() {
15         return instance;
16     }
17
18     public int workerCheck(String userid, String userpw) {
19         String sql = "select pwd from worker where id=?";
20         int result = -1;
21
22         Connection conn = null;
23         PreparedStatement pstmt = null;
24         ResultSet rs = null;
25
26         try {
27             conn = DBManager.getConnection();
28             pstmt = conn.prepareStatement(sql);
29             pstmt.setString(1, userid);
30             rs = pstmt.executeQuery();
31             if (rs.next()) {
32                 result = 0;
33                 String dbPwd = rs.getString(1);
34                 if (dbPwd.equals(userpw)) {
35                     result = 1;
36                 }

```

```

37     }
38     DBManager.close(conn, pstmt, rs);
39     } catch (Exception e) {
40     }
41     return result;
42     }
43 }

```

[설명:시작]

18: 전달인자로 준 아이디와 일치하는 행이 존재하는 관리자 인증을 위한 메소드입니다. 아이디 존재하지 않으면 -1을 아이디 존재하지만 비밀번호가 불일치하면 0을 아이디와 비밀번호 모두 일치하면 1을 결과값으로 리턴합니다.

19: 아이디를 검색 조건으로 주어서 비밀번호를 얻어옵니다.

33: 디비에 저장된 비밀번호를 얻어와 dbPwd 변수에 대입합니다.

34: 디비의 비밀번호와 입력한 비밀번호가 일치하는지 여부를 살펴보고 비밀번호마저도 일치하여 result에 1을 저장합니다.

[설명:끝]

관리자 로그인 페이지에서 아이디와 비밀번호를 입력한 후 [업무 로그인] 버튼을 클릭하면 “NonageServlet?command=admin_login”이 요청됩니다. 로그인 인증 처리를 위한 액션 클래스를 만듭시다.

[직접해보세요] 관리자 로그인 인증 처리를 위한 액션 클래스[파일이름 : srcWcomWnonageWadminWcontrollerWactionWAdminLoginAction.java]

```

1 package com.nonage.admin.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.controller.action.Action;
11 import com.nonage.dao.WorkerDAO;
12
13 public class AdminLoginAction implements Action {
14
15     @Override
16     public void execute(HttpServletRequest request, HttpServletResponse response)
17         throws ServletException, IOException {
18         String url = "NonageServlet?command=admin_login_form";
19         String msg = "";
20         String workerId = request.getParameter("workerId").trim();
21         String workerPwd = request.getParameter("workerPwd").trim();
22

```



```

23 WorkerDAO workerDAO = WorkerDAO.getInstance();
24
25 int result = workerDAO.workerCheck(workerId, workerPwd);
26
27 if (result == 1) { // 로그인 성공
28     HttpSession session = request.getSession();
29     session.setAttribute("workerId", workerId);
30     url = "NonageServlet?command=admin_product_list";
31 } else if (result == 0) {
32     msg = "비밀번호를 확인하세요.";
33 } else if (result == -1) {
34     msg = "아이디를 확인하세요.";
35 }
36 request.setAttribute("message", msg);
37 request.getRequestDispatcher(url).forward(request, response);
38 }
39 }

```

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

```

.. //중복되는 내용 생략
69     } else if (command.equals("admin_login")) {
70         action = new AdminLoginAction();
71     }
.. //중복되는 내용 생략

```

더 이상 로그인한 상태를 유지하지 않으려면 화면 우측의 로그아웃 버튼을 클릭합니다. 그러면 “NonageServlet?command=admin_logout” 요청이 발생합니다. 이 요청을 처리하는 액션 클래스는 다음과 같습니다.

[직접해보세요] 관리자 로그인 인증 처리를 위한 액션 클래스 [파일 이름 :
srcWcomWnonageWadminWcontrollerWactionWAdminLogoutAction.java]

```

1 package com.nonage.admin.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.controller.action.Action;
11
12 public class AdminLogoutAction implements Action {
13
14     @Override

```

```

15 public void execute(HttpServletRequest request, HttpServletResponse response)
16     throws ServletException, IOException {
17     String url="NonageServlet?command=admin_login_form";
18
19     HttpSession session=request.getSession(false);
20     if(session!=null){
21         session.invalidate();
22         request.setAttribute("message", "");
23     }
24
25     request.getRequestDispatcher(url).forward(request, response);
26 }
27 }

```

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

```

.. //중복되는 내용 생략
71     } else if (command.equals("admin_logout")) {
72         action = new AdminLogoutAction();
73     }
.. //중복되는 내용 생략

```

14-2 상품 관리

로그인이 성공적이었다면 관리자 페이지로 이동하게 되는데 관리자 페이지들은 화면 상단과 하단은 물론 좌측에 공통된 메뉴를 제공받습니다. 이를 위한 파일들을 살펴보도록 합니다.

[직접해보세요] 각 화면에서 공통으로 사용되는 헤더 파일을 작성하기[파일이름 : WebContent/admin/header.jsp]

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
4 <%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
5 <!DOCTYPE html>
6 <html>
7 <head>
8 <meta charset="UTF-8">
9 <title>Nonage Admin</title>
10 <link rel="stylesheet" href="admin/css/admin.css">
11 <script type="text/javascript" src="admin/product/product.js"></script>
12 <c:choose>
13   <c:when test="${empty workerId}">
14     <script type="text/javascript">
15       location.href = 'NonageServlet?command=admin_login_form';
16     </script>
17   </c:when>
18 </c:choose>
19 </head>
20 <body onload="go_ab()">
21   <div id="wrap">
22     <header>
23       <div id="logo">
24         <a href="NonageServlet?command=admin_login_form">
25           
26           
27         </a>
28       </div>
29       <input class="btn" type="button" value="logout" style="float: right;"
30         onClick="location.href='NonageServlet?command=admin_logout'">
31     </header>
32   <div class="clear"></div>
```

[직접해보세요] 각 화면에서 공통으로 사용되는 풋터 파일을 작성하기[파일이름 : WebContent/admin/footer.jsp]

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <div class="clear"></div>
4
5 <footer>
6   <hr>
```

```

7      <div id="copy">
8          All contents Copyright 2013 Nonage Inc. all rights reserved<br>
9          Contact mail : Nonage@Nonage.com Tel: +82 64 123 4315
10         Fax +82 64 123 4321
11     </div>
12 </footer>
13 </div>
14 </body>
15 </html>

```

[직접해보세요] 서버 페이지에서 좌측 메뉴를 구성하는 파일[파일이름 : WebContent/admin/sub_menu.jsp]

```

1  <%@ page language="java" contentType="text/html; charset=UTF-8"
2      pageEncoding="UTF-8"%>
3
4  <nav id="sub_menu">
5      <h1>Admin Setting</h1>
6      <ul>
7          <li><a href='NonageServlet?command=admin_product_list'> 상품리스트</a></li>
8          <li><a href='NonageServlet?command=admin_order_list'> 주문리스트</a></li>
9          <li><a href='NonageServlet?command=admin_member_list'> 회원리스트</a></li>
10         <li><a href='NonageServlet?command=admin_qna_list'> Q&A리스트</a></li>
11     </ul>
12 </nav>

```

관리자로 로그인에 성공하였을 경우 제공되는 첫 페이지가 상품 리스트입니다. 이를 위해서는 ProductDAO에 관리자를 위한 상품 처리 메소드를 추가합니다.

[직접해보세요] 상품 테이블을 액세스하는 DAO 클래스에 메소드 추가하기[파일 이름 :srcWcomWnonageWdaoWProductDAO.java]

```

143 //중복되는 내용이어서 생략
144 /*
145  * 관리자 모드에서 사용되는 메소드
146  */
147 public int totalRecord(String product_name) {
148     int total_pages = 0;
149     String sql = "select count(*) from product where name like '%'||?||'%'";
150
151     Connection con = null;
152     PreparedStatement pstmt = null;
153     ResultSet pageset = null;
154
155     try {
156         con = DBManager.getConnection();
157         pstmt = con.prepareStatement(sql);
158

```

```

159     if (product_name.equals("")){
160         pstmt.setString(1, "%");
161     }
162     else{
163         pstmt.setString(1, product_name);
164     }
165     pageset = pstmt.executeQuery();
166
167     if (pageset.next()) {
168         total_pages = pageset.getInt(1); // 레코드의 개수
169         pageset.close();
170     }
171 } catch (Exception e) {
172     e.printStackTrace();
173 } finally {
174     DBManager.close(con, pstmt);
175 }
176 return total_pages;
177 }
178
179 static int view_rows = 5; // 페이지의 개수
180 static int counts = 5; // 한 페이지에 나타낼 상품의 개수
181
182 // 페이지 이동을 위한 메소드
183 public String pageNumber(int tpage, String name) {
184     String str = "";
185
186     int total_pages = totalRecord(name);
187     int page_count = total_pages / counts + 1;
188
189     if (total_pages % counts == 0) {
190         page_count--;
191     }
192     if (tpage < 1) {
193         tpage = 1;
194     }
195
196     int start_page = tpage - (tpage % view_rows) + 1;
197     int end_page = start_page + (counts-1);
198
199     if (end_page > page_count) {
200         end_page = page_count;
201     }
202     if (start_page > view_rows) {
203         str += "<a href='NonageServlet?command=admin_product_list&tpage=1&key="
204             + name + "'>&lt;&lt;</a>&nbsp;&nbsp;&nbsp;";
205         str+="<a href='NonageServlet?command=admin_product_list&tpage="
206             + (start_page - 1);

```

```

207     str += "&key=<%=product_name%>'>&lt;/a>&nbsp;&nbsp;&nbsp;";
208 }
209
210 for (int i = start_page; i <= end_page; i++) {
211     if (i == tpage) {
212         str += "<font color=red>[" + i + "]"&nbsp;&nbsp;&nbsp;</font>";
213     } else {
214         str += "<a href='NonageServlet?command=admin_product_list&tpage="
215             + i + "&key=" + name + "'>[" + i + "]"</a>&nbsp;&nbsp;&nbsp;";
216     }
217 }
218
219 if (page_count > end_page) {
220     str += "<a href='NonageServlet?command=admin_product_list&tpage="
221         + (end_page + 1) + "&key=" + name + "'> &gt; </a>&nbsp;&nbsp;&nbsp;";
222     str += "<a href='NonageServlet?command=admin_product_list&tpage="
223         + page_count + "&key=" + name + "'> &gt; &gt; </a>&nbsp;&nbsp;&nbsp;";
224 }
225 return str;
226 }
227
228 public ArrayList<ProductVO> listProduct(int tpage, String product_name) {
229     ArrayList<ProductVO> productList = new ArrayList<ProductVO>();
230
231     String str = "select pseq, indatex, name, price1, price2, useyn, bestyn " +
232         " from product where name like '%" + product_name + "%' order by pseq desc";
233
234     Connection con = null;
235     PreparedStatement pstmt = null;
236     ResultSet rs = null;
237
238     int absolutePage = 1;
239
240     try {
241         con = DBManager.getConnection();
242         absolutePage = (tpage - 1) * counts + 1;
243         pstmt = con.prepareStatement(str, ResultSet.TYPE_SCROLL_SENSITIVE,
244             ResultSet.CONCUR_UPDATABLE);
245
246         if (product_name.equals("")){
247             pstmt.setString(1, "%");
248         } else{
249             pstmt.setString(1, product_name);
250         }
251
252         rs = pstmt.executeQuery();
253         if (rs.next()) {

```

```

254     rs.absolute(absolutepage);
255     int count = 0;
256
257     while (count < counts) {
258         ProductVO product = new ProductVO();
259         product.setPseq(rs.getInt(1));
260         product.setIndate(rs.getTimestamp(2));
261         product.setName(rs.getString(3));
262         product.setPrice1(rs.getInt(4));
263         product.setPrice2(rs.getInt(5));
264         product.setUseyn(rs.getString(6));
265         product.setBestyn(rs.getString(7));
266         productList.add(product);
267         if (rs.isLast()){
268             break;
269         }
270         rs.next();
271         count++;
272     }
273 }
274 } catch (Exception e) {
275     e.printStackTrace();
276 } finally {
277     DBManager.close(con, pstmt, rs);
278 }
279 return productList;
280 }
281
282 public int insertProduct(ProductVO product) {
283     int result = 0;
284
285     String sql = "insert into product (" +
286         "pseq, kind, name, price1, price2, price3, content, image) " +
287         "values(product_seq.nextval, ?, ?, ?, ?, ?, ?, ?)";
288
289     Connection con = null;
290     PreparedStatement pstmt = null;
291
292     try {
293         con = DBManager.getConnection();
294         pstmt = con.prepareStatement(sql);
295         pstmt.setString(1, product.getKind());
296         pstmt.setString(2, product.getName());
297         pstmt.setInt(3, product.getPrice1());
298         pstmt.setInt(4, product.getPrice2());
299         pstmt.setInt(5, product.getPrice3());
300         pstmt.setString(6, product.getContent());
301         pstmt.setString(7, product.getImage());

```

```

302     result = pstmt.executeUpdate();
303 } catch (Exception e) {
304     System.out.println("추가 실패");
305     e.printStackTrace();
306 } finally {
307     DBManager.close(con, pstmt);
308 }
309 return result;
310 }
311
312 public int updateProduct(ProductVO product) {
313     int result = -1;
314     String sql = "update product set kind=?, useyn=?, name=?" +
315         ", price1=?, price2=?, price3=?, content=?, image=?, bestyn=? " +
316         "where pseq=?";
317
318     Connection con = null;
319     PreparedStatement pstmt = null;
320
321     try {
322         con = DBManager.getConnection();
323         pstmt = con.prepareStatement(sql);
324         pstmt.setString(1, product.getKind());
325         pstmt.setString(2, product.getUseyn());
326         pstmt.setString(3, product.getName());
327         pstmt.setInt(4, product.getPrice1());
328         pstmt.setInt(5, product.getPrice2());
329         pstmt.setInt(6, product.getPrice3());
330         pstmt.setString(7, product.getContent());
331         pstmt.setString(8, product.getImage());
332         pstmt.setString(9, product.getBestyn());
333         pstmt.setInt(10, product.getPseq());
334         result = pstmt.executeUpdate();
335     } catch (Exception e) {
336         e.printStackTrace();
337     } finally {
338         DBManager.close(con, pstmt);
339     }
340     return result;
341 }
342 }

```

상품 관리를 위한 폼에 입력된 정보가 올바른지 판단하는 자바스크립트 파일을 WebContent의 하위 폴더인 product에 product.js란 이름으로 작성합니다. 자바스크립트 파일은 이미 header.jsp 페이지에서 포함하고 있습니다.

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


```
1 function go_cart() {
2     if (document.formm.quantity.value == "") {
3         alert("수량을입력하여주세요.");
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     for ( var i = 0; i < document.formm.cseq.length; i++) {
14         if (document.formm.cseq[i].checked == true) {
15             count++;
16         }
17     }
18     if (count == 0) {
19         alert("삭제할항목을선택해주세요.");
20
21     } else {
22         document.formm.action = "NonageServlet?command=cart_delete"
23         document.formm.submit();
24     }
25 }
26
27 function go_order_insert() {
28     document.formm.action = "NonageServlet?command=order_insert"
29     document.formm.submit();
30 }
31
32 function go_order_delete() {
33     var count = 0;
34     for ( var i = 0; i < document.formm.oseq.length; i++) {
35         if (document.formm.oseq[i].checked == true) {
36             count++;
37         }
38     }
39     if (count == 0) {
40         alert("삭제할항목을선택해주세요.");
41
42     } else {
43         document.formm.action = "NonageServlet?command=order_delete"
44         document.formm.submit();
45     }
46 }
47
```

```

48 function go_order() {
49     document.formm.action = "NonageServlet?command=mypage"
50     document.formm.submit();
51 }

```

2.1 상품 리스트

관리자 로그인 페이지에서 아이디와 비밀번호를 입력받아 [업무 로그인] 버튼을 클릭하여 로그인이 성공적이었다면 "NonageServlet?command=admin_product_list" 요청이 발생합니다. 이 요청은 상품 리스트를 출력하는 페이지를 요청합니다. 이 요청을 받아 처리하는 액션을 만들어 봅시다.

[직접해보세요] 상품 리스트 페이지를 출력하는 액션 클래스[파일이름 : srcWcomWnonageWadminWcontrollerWactionWAdminProductListAction.java]

```

1 package com.nonage.admin.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
10 import com.nonage.controller.action.Action;
11 import com.nonage.dao.ProductDAO;
12 import com.nonage.dto.ProductVO;
13
14 public class AdminProductListAction implements Action {
15
16     @Override
17     public void execute(HttpServletRequest request, HttpServletResponse response)
18         throws ServletException, IOException {
19
20         String url = "admin/product/productList.jsp";
21
22         String key=request.getParameter("key");
23         String tpage=request.getParameter("tpage");
24         if(key==null){
25             key="";
26         }
27         if(tpage== null){
28             tpage="1"; //현재 페이지 (default 1)
29         }else if(tpage.equals("")){
30             tpage="1";
31         }
32         request.setAttribute("key", key);
33         request.setAttribute("tpage",tpage);
34

```

```

35     ProductDAO productDAO=ProductDAO.getInstance();
36     ArrayList<ProductVO> productList=
37         productDAO.listProduct(Integer.parseInt(tpage), key);
38
39     String paging=productDAO.pageNumber(Integer.parseInt(tpage), key);
40
41     request.setAttribute("productList",productList);
42     int n=productList.size();
43     request.setAttribute("productListSize",n);
44     request.setAttribute("paging", paging);
45     request.getRequestDispatcher(url).forward(request, response);
46 }
47 }

```

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

```

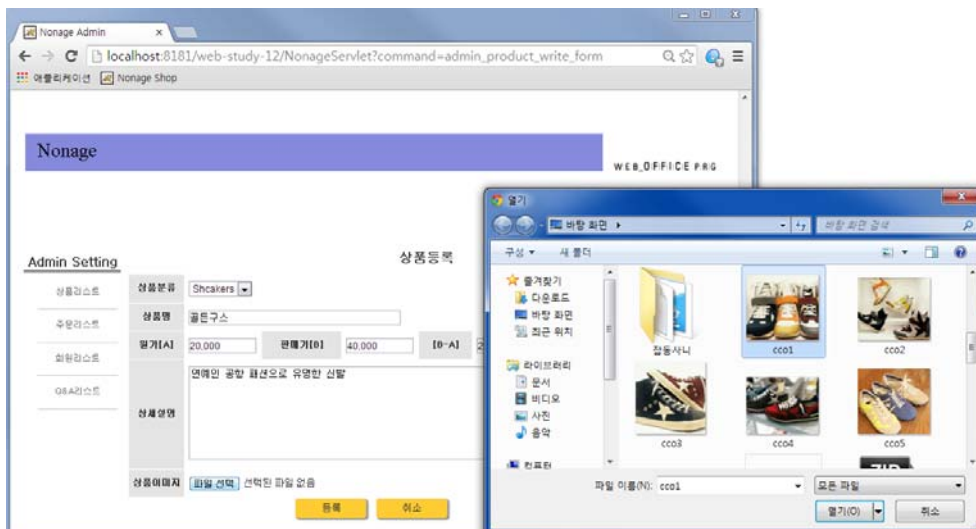
.. //중복되는 내용 생략
73     } else if (command.equals("admin_product_list")) {
74         action = new AdminProductListAction();
75     }
.. //중복되는 내용 생략

```

상품 리스트 화면에서는 상품 등록, 상품 상세 보기, 상품 수정 처리를 실행할 수 있습니다.

▼ 상품 등록 화면

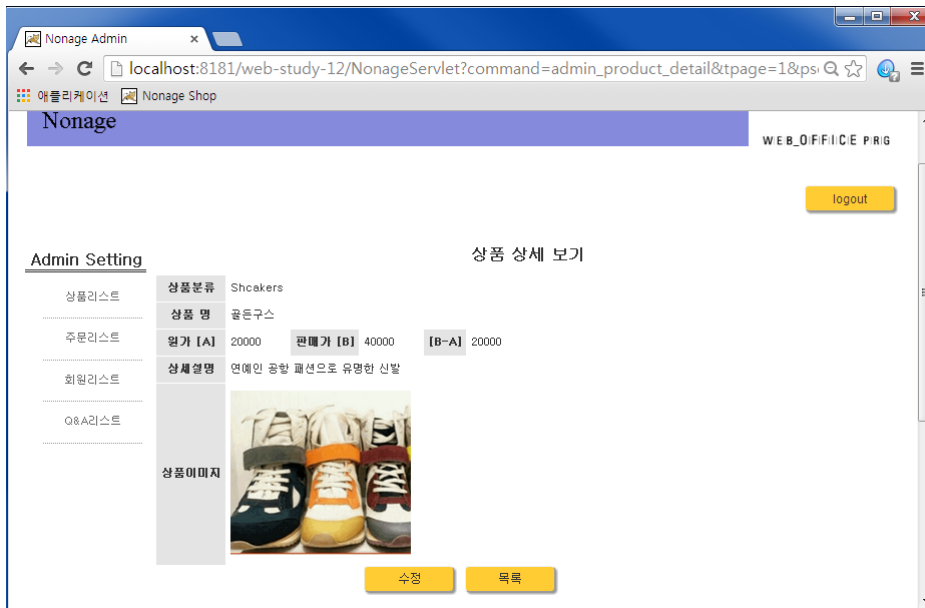
상품 리스트 화면에서 [상품 등록] 버튼을 클릭하면 상품 등록 화면으로 이동합니다.



상품 정보를 입력하고 [등록] 버튼을 클릭하면 상품 등록 처리가 완료되어 상품 리스트 화면으로 이동합니다.

▼ 상품 상세보기

상품 상세보기 페이지는 일반 게시판 글 내용 보기 부분과 동일한 코드로 되어 있습니다.
 일반 게시판 코드 부분에 이미지를 출력할 수 있도록 코드를 수정해서 사용합니다.



▼ 상품 수정 화면

상품의 정보를 수정하는 페이지입니다. 상품 삭제는 따로 하지 않고 삭제 대신 사용의무 칼럼을 두어 이를 제어해서 상품 판매가 불가능하도록 합니다.



2.2 상품 등록

상품 등록은 게시판을 응용한 대표적인 경우입니다. 상품 등록이 어떻게 구축되는지에 대한 전체 흐름을 보여주는 예제가 있는데, 이 예제를 기반으로 실제로 쓸 수 있는 코드로 바꿔 보도록 합니다.

productList.jsp 페이지는 상품이 등록된 페이지이며, 이곳에서 [상품등록]을 클릭하면 "NonageServlet?command=admin_product_write_form"가 요청되어 상품 등록 페이지로 이

동합니다. 상품을 등록하기 위해서 이미지가 필요한데 이미지를 product_images 디렉터리에 저장하기 때문에 product_images 디렉터리를 미리 만들어 두어야 합니다(이미 쇼핑몰 사용자 페이지를 만들면서 만들어 두었다). 상품 등록을 위해 상품 정보를 입력받는 입력 폼을 출력하는 액션 클래스를 만듭시다.

[직접해보세요] 상품 정보를 입력받는 폼을 출력하는 액션 클래스[파일이름 : srcWcomWnonageWadminWcontrollerWactionWAdminProductWriteFormAction.java]

```

1 package com.nonage.admin.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.controller.action.Action;
10
11 public class AdminProductWriteFormAction implements Action {
12
13     @Override
14     public void execute(HttpServletRequest request, HttpServletResponse response)
15         throws ServletException, IOException {
16
17         String url = "admin/product/productWrite.jsp";
18         String kindList[] = { "Heels", "Boots", "Sandals", "Slippers",
19             "Shcakers", "Sale" };
20         request.setAttribute("kindList", kindList);
21         request.getRequestDispatcher(url).forward(request, response);
22     }
23 }

```

[직접해보세요] 상품 정보를 입력받는 JSP 페이지[파일이름 : WebContent/admin/product/productWrite.jsp]

```

1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3 <%@ include file="/admin/header.jsp"%>
4 <%@ include file="/admin/sub_menu.jsp"%>
5
6 <article>
7 <h1>상품등록</h1>
8 <form name="frm" method="post" enctype="multipart/form-data">
9 <table id="list">
10 <tr>
11 <th>상품분류</th>
12 <td colspan="5">
13 <select name="kind">
14 <c:forEach items="${kindList}" var="kind" varStatus="status">
15 <option value="${status.count}">${kind}</option>

```

```

16     </c:forEach>
17 </select>
18 <tr>
19     <th>상품명</th>
20     <td width="343" colspan="5">
21         <input type="text" name="name" size="47" maxlength="100" >
22     </td>
23 </tr>
24 <tr>
25     <th>원가[A]</th>
26     <td width="70">
27         <input type="text" name="price1" size="11" onKeyUp='NumFormat(this)'\>
28     </td>
29     <th>판매가[B]</th>
30     <td width="70">
31         <input type="text" name="price2" size="11" onBlur="go_ab()"
32     onKeyUp='NumFormat(this)'\>
33     </td>
34     <th>[B-A]</th>
35     <td width="72">
36         <input type="text" name="price3" size="11" readonly
37     onKeyUp='NumFormat(this)'\>
38     </td>
39 </tr>
40
41 <tr>
42     <th>상세설명</th>
43     <td colspan="5">
44         <textarea name="content" rows="8" cols="70" ></textarea>
45     </td>
46 </tr>
47 <tr>
48     <th>상품이미지</th>
49     <td width="343" colspan="5">
50         <input type="file" name="image">
51     </td>
52 </tr>
53 </table>
54 <input class="btn" type="button" value="등록" onClick="go_save()">
55 <input class="btn" type="button" value="취소" onClick="go_mov()">
56 </form>
57 </article>
58 <%@ include file="/admin/footer.jsp"%>
59 </body>
60 </html>

```

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

```

.. //중복되는 내용 생략
75     } else if (command.equals("admin_product_write_form")) {
76         action = new AdminProductWriteFormAction();
77     }
.. //중복되는 내용 생략

```

상품 정보를 입력한 후에 [등록] 버튼을 클릭하면 “NonageServlet?command=admin_product_write” 요청이 발생하여 데이터베이스에 상품 정보를 추가하게 됩니다. 이 요청을 처리할 액션 클래스를 만듭시다.

[직접해보세요] 상품 정보를 데이터베이스에 추가하는 액션 클래스[파일이름 : srcWcomWnonageWadminWcontrollerWactionWAdminProductWriteAction.java]

```

1 package com.nonage.admin.controller.action;
2
3 import java.io.IOException;
4
5 import javax.servlet.ServletContext;
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.controller.action.Action;
12 import com.nonage.dao.ProductDAO;
13 import com.nonage.dto.ProductVO;
14 import com.oreilly.servlet.MultipartRequest;
15 import com.oreilly.servlet.multipart.DefaultFileRenamePolicy;
16
17 public class AdminProductWriteAction implements Action {
18
19     @Override
20     public void execute(HttpServletRequest request, HttpServletResponse response)
21         throws ServletException, IOException {
22         String url = "NonageServlet?command=admin_product_list";
23
24         HttpSession session = request.getSession();
25
26         int sizeLimit = 5 * 1024 * 1024;
27         String savePath = "product_images";
28         ServletContext context = session.getServletContext();
29         String uploadFilePath = context.getRealPath(savePath);
30
31         MultipartRequest multi = new MultipartRequest(request, // 1. 요청 객체
32             uploadFilePath, // 2. 업로드될 파일이 저장될 파일 경로명
33             sizeLimit, // 3. 업로드될 파일의 최대 크기(5Mb)
34             "UTF-8", // 4. 인코딩 타입 지정
35             new DefaultFileRenamePolicy() // 5. 덮어쓰기를 방지 위한 부분

```

```

36     ); // 이 시점을 기해 파일은 이미 저장에 되었다
37
38     ProductVO productVO = new ProductVO();
39     productVO.setKind(multi.getParameter("kind"));
40     productVO.setName(multi.getParameter("name"));
41     productVO.setPrice1(Integer.parseInt(multi.getParameter("price1")));
42     productVO.setPrice2(Integer.parseInt(multi.getParameter("price2")));
43     productVO.setPrice3(Integer.parseInt(multi.getParameter("price2"))
44         - Integer.parseInt(multi.getParameter("price1")));
45     productVO.setContent(multi.getParameter("content"));
46     productVO.setImage(multi.getFilesystemName("image"));
47
48     ProductDAO productDAO = ProductDAO.getInstance();
49     productDAO.insertProduct(productVO);
50
51     response.sendRedirect(url);
52 }
53 }

```

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

```

.. //중복되는 내용 생략
77     } else if (command.equals("admin_product_write")) {
78         action = new AdminProductWriteAction();
79     }
.. //중복되는 내용 생략

```

상품 리스트에서 상품명을 클릭하면 “NonageServlet?command=admin_product_detail” 요청이 발생하여 상품 상세보기 페이지로 들어가게 됩니다. 이 요청을 처리할 액션 클래스를 만듭시다.

[직접해보세요] 상품 상세 정보를 보여줄 액션 클래스 [파일 이름 :
srcWcomWnonageWadminWcontrollerWactionWAdminProductDetailAction.java]

```

1  package com.nonage.admin.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.ProductDAO;
10 import com.nonage.controller.action.Action;
11 import com.nonage.dto.ProductVO;
12
13 public class AdminProductDetailAction implements Action {
14

```



```

15  @Override
16  public void execute(HttpServletRequest request, HttpServletResponse response)
17      throws ServletException, IOException {
18      String url = "admin/product/productDetail.jsp";
19
20      String pseq = request.getParameter("pseq").trim();
21
22      ProductDAO productDAO = ProductDAO.getInstance();
23      ProductVO productVO = productDAO.getProduct(pseq);
24
25      request.setAttribute("productVO", productVO);
26
27      String tpage = "1";
28      if (request.getParameter("tpage") != null) {
29          tpage = request.getParameter("tpage");
30      }
31      String kindList[] = { "0", "Heels", "Boots", "Sandals", "Slippers",
32          "Shcakers", "Sale" };
33      request.setAttribute("tpage", tpage);
34      int index=Integer.parseInt(productVO.getKind().trim());
35      request.setAttribute("kind", kindList[index]);
36
37      request.getRequestDispatcher(url).forward(request, response);
38  }
39  }

```

[직접해보세요] 상품 상세 정보를 보여주는 JSP 페이지 [파일이름 : WebContent/admin/product/productDetail.jsp]

```

1  <%@ page language="java" contentType="text/html; charset=UTF-8"
2      pageEncoding="UTF-8"%>
3  <%@ include file="/admin/header.jsp"%>
4  <%@ include file="/admin/sub_menu.jsp"%>
5
6  <article>
7  <h1>상품 상세 보기</h1>
8  <form name="frm" method="post">
9  <table id="list">
10     <tr>
11         <th>상품분류</th>
12         <td colspan="5">
13             ${kind}
14         </td>
15     </tr>
16     <tr>
17         <th align="center" >상품 명</th>
18         <td colspan="5">${productVO.name}</td>
19     </tr>
20     <tr>

```

```

21      <th>원가 [A]</th><td width="60">${productVO.price1}</td>
22      <th>판매가 [B]</th><td width="60">${productVO.price2}</td>
23      <th>[B-A]</th><td>${productVO.price3}</td>
24  </tr>
25
26  <tr>
27      <th>상세설명</th><td colspan="5">${productVO.content}</td>
28  </tr>
29  <tr>
30      <th>상품이미지</th>
31      <td colspan="5" align="center">
32          
33      </td>
34  </tr>
35 </table>
36 <input class="btn" type="button" value="수정"
37     onClick="go_mod('${tpage}','${productVO.pseq}')">
38 <input class="btn" type="button" value="목록" onClick="go_list('${tpage}')">
39 </form>
40 </article>
41 <%@ include file="/admin/footer.jsp"%>
42 </body>
43 </html>

```

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

```

.. //중복되는 내용 생략
79     } else if (command.equals("admin_product_detail")) {
80         action = new AdminProductDetailAction();
81     }
.. //중복되는 내용 생략

```

상품 상세 보기에서 [수정] 버튼을 클릭하면
“NonageServlet?command=admin_product_update_form” 요청이 발생하여 상품 정보를 수정할 수 있도록 이미 저장된 정보를 한번 보여주고 수정된 정보를 입력받아 갱신하도록 하는 페이지로 이동합니다. 이 요청을 처리할 액션 클래스를 만듭시다.

[직접해보세요] 수정할 상품 정보를 입력받는 폼을 출력하는 액션 클래스 [파일이름 :
srcWcomWnonageWadminWcontrollerWactionWAdminProductUpdateFormAction.java]

```

1 package com.nonage.admin.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.ProductDAO;
10 import com.nonage.controller.action.Action;
11 import com.nonage.dto.ProductVO;
12
13 public class AdminProductUpdateFormAction implements Action {
14
15     @Override
16     public void execute(HttpServletRequest request, HttpServletResponse response)
17         throws ServletException, IOException {
18         String url = "admin/product/productUpdate.jsp";
19
20         String pseq = request.getParameter("pseq").trim();
21
22         ProductDAO productDAO = ProductDAO.getInstance();
23         ProductVO productVO = productDAO.getProduct(pseq);
24
25         request.setAttribute("productVO", productVO);
26
27         String tpage = "1";
28         if (request.getParameter("tpage") != null) {
29             tpage = request.getParameter("tpage");
30         }
31         request.setAttribute("tpage", tpage);
32
33         String kindList[] = { "Heels", "Boots", "Sandals", "Slippers",
34             "Shcakers", "Sale" };
35         request.setAttribute("kindList", kindList);
36
37         request.getRequestDispatcher(url).forward(request, response);
38     }
39 }

```

[직접해보세요] 상품 정보 수정을 위한 JSP 페이지 [파일 이름 : WebContent/admin/product/productUpdate.jsp]

```

1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <%@ include file="/admin/header.jsp"%>
4 <%@ include file="/admin/sub_menu.jsp"%>
5
6 <article>
7 <h1>상품수정</h1>
8 <form name="frm" method="post" enctype="multipart/form-data">
9 <input type="hidden" name="pseq" value="${productVO.pseq}">
10 <input type="hidden" name="code" >
11 <input type="hidden" name="nonmakeImg" value="${productVO.image}">
12 <table id="list">
13 <tr>

```

```

14     <th>상품분류</th>
15     <td colspan="5">
16         <select name="kind">
17             <c:forEach items="${kindList}" var="kind" varStatus="status">
18                 <c:choose>
19                     <c:when test="${productVO.kind==status.count}">
20                         <option value="${status.count}" selected="selected">${kind}</option>
21                     </c:when>
22                     <c:otherwise>
23                         <option value="${status.count}">${kind}</option>
24                     </c:otherwise>
25                 </c:choose>
26             </c:forEach>
27         </select>
28     </td>
29 </tr>
30 <tr>
31     <th>상품명</th>
32     <td width="343" colspan="5">
33         <input type="text" name="name" size="47" maxlength="100"
34 value="${productVO.name}">
35     </td>
36 </tr>
37 <tr>
38     <th>원가[A]</th>
39     <td width="70">
40         <input type="text" name="price1" size="11" onKeyUp='NumFormat(this)'
41 value="${productVO.price1}">
42     </td>
43     <th>판매가[B]</th>
44     <td width="70">
45         <input type="text" name="price2" size="11" onBlur="go_ab()"
46 onKeyUp='NumFormat(this)' value="${productVO.price2}">
47     </td>
48     <th>[B-A]</th>
49     <td width="72">
50         <input type="text" name="price3" size="11" readonly
51 onKeyUp='NumFormat(this)'>
52     </td>
53 </tr>
54 <tr>
55     <th>베스트상품</th>
56     <td>
57         <c:choose>
58             <c:when test='${productVO.bestyn=="y"}'>
59                 <input type="checkbox" name="bestyn" value="y" checked="checked">
60             </c:when>
61             <c:otherwise>

```

```

62         <input type="checkbox" name="bestyn" value="n">
63         </c:otherwise>
64     </c:choose>
65 </td>
66 <th>사용유무</th>
67 <td>
68     <c:choose>
69         <c:when test='${productVO.useyn=="y"}'>
70             <input type="checkbox" name="useyn" value="y" checked="checked">
71         </c:when>
72         <c:otherwise>
73             <input type="checkbox" name="useyn" value="n">
74         </c:otherwise>
75     </c:choose>
76 </td>
77 </tr>
78 <tr>
79     <th>상세설명</th>
80     <td colspan="5">
81         <textarea name="content" rows="8" cols="70" >
82             ${productVO.content}
83         </textarea>
84     </td>
85 </tr>
86 <tr>
87     <th>상품이미지</th>
88     <td colspan="5">
89         
90         <br>
91         <input type="file" name="image">
92     </td>
93 </tr>
94 </table>
95 <input class="btn" type="button" value="수정"
96     onClick="go_mod_save('${tpage}','${productVO.pseq}')">
97 <input class="btn" type="button" value="취소" onClick="go_mov()">
98 </form>
99 </article>
100 <%@ include file="/admin/footer.jsp"%>
101 </body>
102 </html>

```

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

```

.. //중복되는 내용 생략
81     } else if (command.equals("admin_product_update_form")) {
82         action = new AdminProductUpdateFormAction();
83     }

```

상품 정보를 수정하고 [수정] 버튼을 클릭하면 데이터베이스 상품 정보를 수정하기 위한 "NonageServlet?command=admin_product_update" 요청이 발생합니다. 이 요청을 받아 처리하는 액션 클래스를 만들어 봅시다.

[직접해보세요] 상품 정보를 데이터베이스에 수정하는 액션 클래스[파일이름] : srcWcomWnonageWadminWcontrollerWactionWAdminProductUpdateAction.java]

```

1 package com.nonage.admin.controller.action;
2
3 import java.io.IOException;
4
5 import javax.servlet.ServletContext;
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.controller.action.Action;
12 import com.nonage.dao.ProductDAO;
13 import com.nonage.dto.ProductVO;
14 import com.oreilly.servlet.MultipartRequest;
15 import com.oreilly.servlet.multipart.DefaultFileRenamePolicy;
16
17 public class AdminProductUpdateAction implements Action {
18
19     @Override
20     public void execute(HttpServletRequest request, HttpServletResponse response)
21         throws ServletException, IOException {
22         String url = "NonageServlet?command=admin_product_list";
23
24         HttpSession session = request.getSession();
25
26         int sizeLimit = 5 * 1024 * 1024;
27         String savePath = "product_images";
28         ServletContext context = session.getServletContext();
29         String uploadFilePath = context.getRealPath(savePath);
30
31         MultipartRequest multi = new MultipartRequest(request, // 1. 요청 객체
32             uploadFilePath, // 2. 업로드될 파일이 저장될 파일 경로명
33             sizeLimit, // 3. 업로드될 파일의 최대 크기(5Mb)
34             "UTF-8", // 4. 인코딩 타입 지정
35             new DefaultFileRenamePolicy() // 5. 덮어쓰기를 방지 위한 부분
36         ); // 이 시점을 기해 파일은 이미 저장에 되었다
37
38         ProductVO productVO = new ProductVO();
39         productVO.setPseq(Integer.parseInt(multi.getParameter("pseq")));

```

```

40    productVO.setKind(multi.getParameter("kind"));
41    productVO.setName(multi.getParameter("name"));
42    productVO.setPrice1(Integer.parseInt(multi.getParameter("price1")));
43    productVO.setPrice2(Integer.parseInt(multi.getParameter("price2")));
44    productVO.setPrice3(Integer.parseInt(multi.getParameter("price2"))
45        - Integer.parseInt(multi.getParameter("price1")));
46    productVO.setContent(multi.getParameter("content"));
47    if(multi.getFilesystemName("image")==null){
48        productVO.setImage(multi.getParameter("nonmakeimg"));
49    }else{
50        productVO.setImage(multi.getFilesystemName("image"));
51    }
52
53    ProductDAO productDAO = ProductDAO.getInstance();
54    productDAO.updateProduct(productVO);
55
56    response.sendRedirect(url);
57 }
58 }

```

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

```

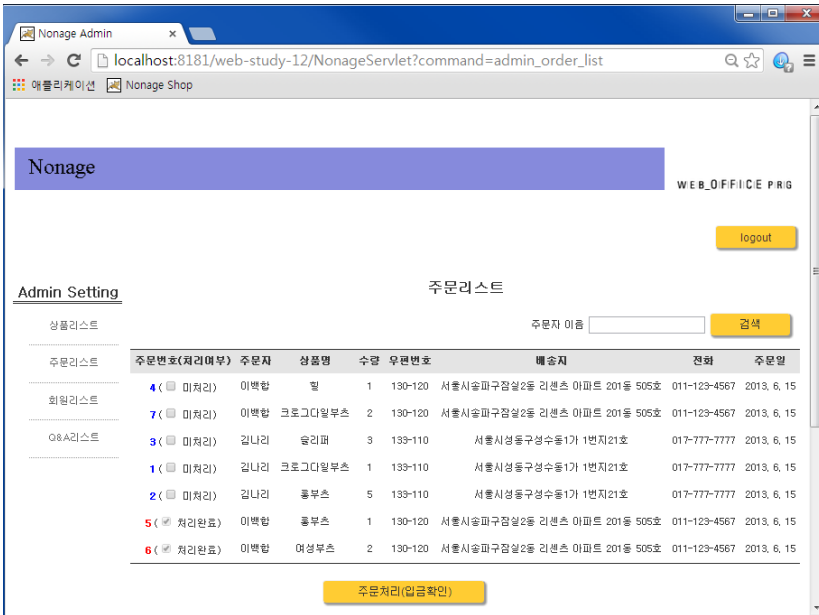
.. //중복되는 내용 생략
83     } else if (command.equals("admin_product_update")) {
84         action = new AdminProductUpdateAction();
85     }
.. //중복되는 내용 생략

```

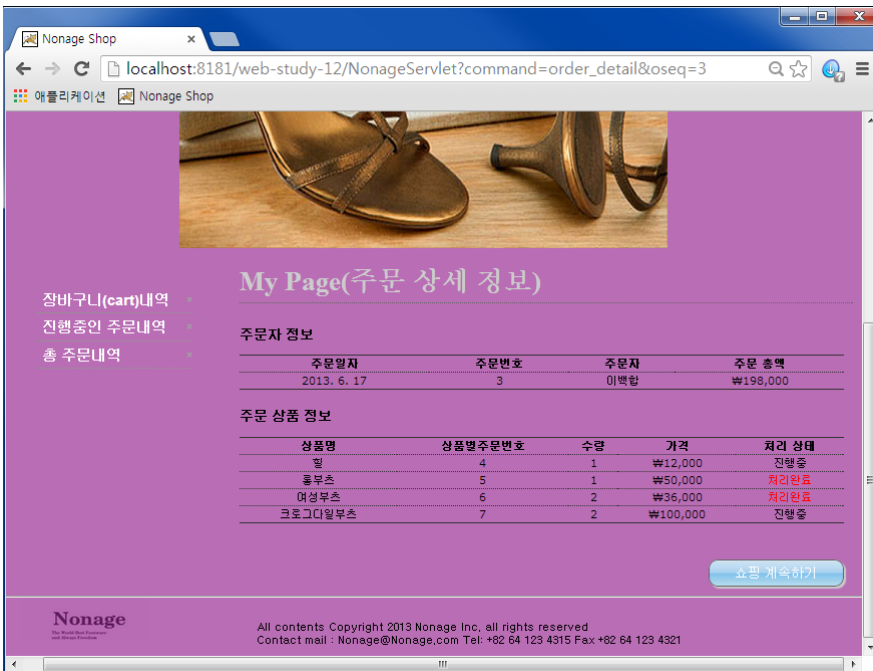
상품 정보를 삭제하는 처리는 없고 삭제하는 대신 판매가 불가능한 상품은 사용유무 컬럼을
 체크하여 처리합니다.

14-3 주문관리

상품 관리가 끝났다면 주문 관리를 하도록 합시다. 회원이 주문 후 무통장 입금을 한 것이 확인되었다면 상품을 배송한 후 주문 처리를 완료해야 합니다.



관리자가 주문처리를 완료하면 사용자 페이지도 주문 처리된 것으로 내용이 바뀌어 나타납니다.



주문 처리를 위한 DAO에 관리자가 주문 처리를 하기 위해 필요한 메소드를 추가합니다.

[직접해보세요] 주문 테이블을 액세스하는 DAO 클래스에 메소드 추가하기[파일 이름 :srcWcomWnonageWdaoWProductDAO.java]


```

151 //중복되는 내용이어서 생략
152 /* *
153  * 관리자 모드에서 사용되는 메소드 * *
154 */
155 public ArrayList<OrderVO> listOrder(String member_name) {
156     ArrayList<OrderVO> orderList = new ArrayList<OrderVO>();
157     String sql = "select * from order_view where mname like '%'||?||'%' "+
158 " order by result, oseq desc";
159
160     Connection conn = null;
161     PreparedStatement pstmt = null;
162     ResultSet rs = null;
163
164     try {
165         conn = DBManager.getConnection();
166         pstmt = conn.prepareStatement(sql);
167         if (member_name == "") {
168             pstmt.setString(1, "%");
169         } else {
170             pstmt.setString(1, member_name);
171         }
172         rs = pstmt.executeQuery();
173         while (rs.next()) {
174             OrderVO orderVO = new OrderVO();
175             orderVO.setOdseq(rs.getInt("ODSEQ"));
176             orderVO.setOseq(rs.getInt("OSEQ"));
177             orderVO.setId(rs.getString("ID"));
178             orderVO.setPseq(rs.getInt("PSEQ"));
179             orderVO.setMname(rs.getString("MNAME"));
180             orderVO.setPname(rs.getString("PNAME"));
181             orderVO.setQuantity(rs.getInt("QUANTITY"));
182             orderVO.setZipNum(rs.getString("ZIP_NUM"));
183             orderVO.setAddress(rs.getString("ADDRESS"));
184             orderVO.setPhone(rs.getString("PHONE"));
185             orderVO.setIndate(rs.getTimestamp("INDATE"));
186             orderVO.setPrice2(rs.getInt("PRICE2"));
187             orderVO.setResult(rs.getString("RESULT"));
188             orderList.add(orderVO);
189         }
190     } catch (Exception e) {
191         e.printStackTrace();
192     } finally {
193         DBManager.close(conn, pstmt, rs);
194     }
195     return orderList;
196 }
197
198 public void updateOrderResult(String oseq) {

```

```

199     String sql = "update order_detail set result='2' where odseq=?";
200
201     Connection conn = null;
202     PreparedStatement pstmt = null;
203
204     try {
205         conn = DBManager.getConnection();
206         pstmt = conn.prepareStatement(sql);
207         pstmt.setString(1, oseq);
208         pstmt.executeUpdate();
209     } catch (Exception e) {
210         e.printStackTrace();
211     } finally {
212         DBManager.close(conn, pstmt);
213     }
214 }
215 }

```

입금 처리가 확인되면 주문 내역을 확인하여 주문 처리를 하기 위해서는 화면 왼쪽의 서브 메뉴에서 주문 리스트 링크를 클릭합니다. 그러면 “NonageServlet?command=admin_order_list” 요청이 발생합니다. 이 요청을 처리하기 위한 액션 클래스를 만듭시다.

[직접해보세요] 관리자 페이지에서 회원들의 주문 리스트를 출력하기 위한 액션 클래스
[파일이름:srcWcomWnonageWadminWcontrollerWactionWAdminOrderListAction.java]

```

1  package com.nonage.admin.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
10 import com.nonage.controller.action.Action;
11 import com.nonage.dao.OrderDAO;
12 import com.nonage.dto.OrderVO;
13
14 public class AdminOrderListAction implements Action {
15
16     @Override
17     public void execute(HttpServletRequest request, HttpServletResponse response)
18         throws ServletException, IOException {
19
20         String url = "admin/order/orderList.jsp";
21         String key = "";

```

```

22     if (request.getParameter("key") != null) {
23         key = request.getParameter("key");
24     }
25
26     OrderDAO orderDAO = OrderDAO.getInstance();
27     ArrayList<OrderVO> orderList = orderDAO.listOrder(key);
28
29     request.setAttribute("orderList", orderList);
30
31     request.getRequestDispatcher(url).forward(request, response);
32 }
33 }

```

[직접해보세요] 주문 리스트를 출력하는 JSP 페이지

[파일이름 : WebContent/admin /order/orderList.jsp]

```

1  <%@ page language="java" contentType="text/html; charset=UTF-8"
2      pageEncoding="UTF-8"%>
3  <%@ include file="/admin/header.jsp"%>
4  <%@ include file="/admin/sub_menu.jsp"%>
5  <script type="text/javascript">
6      function go_order_save() {
7          var count = 0;
8          if (document.frm.result.length == undefined) {
9              if (document.frm.result.checked == true) {
10                 count++;
11             }
12         } else {
13             for ( var i = 0; i < document.frm.result.length; i++) {
14                 if (document.frm.result[i].checked == true) {
15                     count++;
16                 }
17             }
18         }
19         if (count == 0) {
20             alert("주문처리할 항목을 선택해 주세요.");
21         } else {
22             document.frm.action = "NonageServlet?command=admin_order_save";
23             document.frm.submit();
24         }
25     }
26 </script>
27 <article>
28 <h1>주문리스트</h1>
29 <form name="frm" method="post">
30     <table style="float: right;">
31         <tr>
32             <td>주문자 이름 <input type="text" name="key">

```

```

33      <input class="btn" type="button" value="검색" onclick="go_search()">
34      </td>
35  </tr>
36 </table>
37 <br>
38 <table id="orderList">
39 <tr>
40   <th>주문번호(처리여부)</th><th>주문자</th><th>상품명</th><th>수량</th>
41   <th>우편번호</th><th>배송지</th><th>전화</th><th>주문일</th>
42 </tr>
43 <c:forEach items="${orderList}" var="orderVO">
44 <tr>
45   <td>
46     <c:choose>
47       <c:when test='${orderVO.result=="1"}'>
48         <span style="font-weight: bold; color: blue">${orderVO.odseq}</span>
49         (<input type="checkbox" name="result" value="${orderVO.odseq}"> 미처리)
50       </c:when>
51       <c:otherwise>
52         <span style="font-weight: bold; color: red">${orderVO.odseq}</span>
53         (<input type="checkbox" checked="checked" disabled="disabled">처리완료)
54       </c:otherwise>
55     </c:choose>
56   </td>
57   <td>${orderVO.mname}</td> <td>${orderVO.pname}</td>
58   <td>${orderVO.quantity}</td> <td>${orderVO.zipNum}</td>
59   <td>${orderVO.address}</td> <td>${orderVO.phone}</td>
60   <td><fmt:formatDate value="${orderVO.indate}" /></td>
61 </tr>
62 </c:forEach>
63 </table>
64 <input type="button" class="btn" style="width: 200px"
65   value="주문처리(입금확인)" onClick="go_order_save()">
66 </form>
67 </article>
68 <%@ include file="/admin/footer.jsp"%>
69 </body>
70 </html>

```

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

```

.. //중복되는 내용 생략
85   } else if (command.equals("admin_order_list")) {
86     action = new AdminOrderListAction();
87   }
.. //중복되는 내용 생략

```

주문 리스트에서 주문 처리할 항목에 대해서 체크박스를 체크한 후 [주문처리(입금확인) 버튼]을 클릭하면 주문 처리를 완료시키기 위한 “NonageServlet?command=admin_order_save” 요청이 발생합니다. 이 요청을 받아 처리하는 액션을 만들어 봅시다.

[직접해보세요] 주문 처리를 하기 위한 액션 클래스[파일이름 : srcWcomWnonageWadminWcontrollerWactionWAdminOrderSaveAction.java]

```

1 package com.nonage.admin.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.controller.action.Action;
10 import com.nonage.dao.OrderDAO;
11
12 public class AdminOrderSaveAction implements Action {
13
14     @Override
15     public void execute(HttpServletRequest request, HttpServletResponse response)
16         throws ServletException, IOException {
17         String url = "NonageServlet?command=admin_order_list";
18
19         String[] resultArr = request.getParameterValues("result");
20
21         for(String oseq:resultArr){
22             System.out.println(oseq);
23             OrderDAO orderDAO = OrderDAO.getInstance();
24             orderDAO.updateOrderResult(oseq);
25         }
26         request.getRequestDispatcher(url).forward(request, response);
27     }
28 }

```

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

```

.. //중복되는 내용 생략
87     } else if (command.equals("admin_order_save")) {
88         action = new AdminOrderSaveAction();
89     }
.. //중복되는 내용 생략

```

14-4 회원 관리

이번에는 회원 정보를 보기 위한 회원 리스트 메뉴를 처리해 보도록 합니다. 우선 회원 정보를 처리하는 DAO 클래스에 메소드를 추가합니다.

[직접해보세요] 회원 테이블을 액세스하는 DAO 클래스 만들기

[파일 이름 :srcWcomWnonageWdaoWMemberDAO.java]

```
108 //중복되는 내용 생략
109 /* *
110  * 관리자 모드에서 사용되는 메소드 * *
111  */
112 public ArrayList<MemberVO> listMember(String member_name) {
113     ArrayList<MemberVO> memberList = new ArrayList<MemberVO>();
114     String sql = "select * from member where name like '%'||?||'%' "+
115 " order by indate desc";
116
117     Connection conn = null;
118     PreparedStatement pstmt = null;
119     ResultSet rs = null;
120
121     try {
122         conn = DBManager.getConnection();
123         pstmt = conn.prepareStatement(sql);
124         if (member_name == "") {
125             pstmt.setString(1, "%");
126         } else {
127             pstmt.setString(1, member_name);
128         }
129         rs = pstmt.executeQuery();
130         while (rs.next()) {
131             MemberVO memberVO = new MemberVO();
132             memberVO.setId(rs.getString("id"));
133             memberVO.setPwd(rs.getString("pwd"));
134             memberVO.setName(rs.getString("name"));
135             memberVO.setEmail(rs.getString("email"));
136             memberVO.setZipNum(rs.getString("zip_num"));
137             memberVO.setAddress(rs.getString("address"));
138             memberVO.setPhone(rs.getString("phone"));
139             memberVO.setUseyn(rs.getString("useyn"));
140             memberVO.setIndate(rs.getTimestamp("indate"));
141             memberList.add(memberVO);
142         }
143     } catch (Exception e) {
144         e.printStackTrace();
145     } finally {
146         DBManager.close(conn, pstmt, rs);
147     }
148     return memberList;
```

149	}
150	}

화면 좌측의 회원 리스트를 클릭하면 “NonageServlet?command=admin_member_list” 요청이 발생합니다. 이를 위한 요청 클래스를 만듭시다.

[직접해보세요] 회원 리스트를 출력하기 위한 액션 클래스[파일이름 : srcWcomWnonageWadminWcontrollerWactionWAdminMemberListAction.java]

1	package com.nonage.admin.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	
10	import com.nonage.controller.action.Action;
11	import com.nonage.dao.MemberDAO;
12	import com.nonage.dto.MemberVO;
13	
14	public class AdminMemberListAction implements Action {
15	
16	@Override
17	public void execute(HttpServletRequest request, HttpServletResponse response)
18	throws ServletException, IOException {
19	
20	String url = "admin/member/memberList.jsp";
21	String key = "";
22	if (request.getParameter("key") != null) {
23	key = request.getParameter("key");
24	}
25	
26	MemberDAO memberDAO = MemberDAO.getInstance();
27	ArrayList<MemberVO> memberList = memberDAO.listMember(key);
28	
29	request.setAttribute("memberList", memberList);
30	
31	request.getRequestDispatcher(url).forward(request, response);
32	}
33	}

[직접해보세요] 회원 리스트를 출력하는 JSP 파일
[파일이름 : WebContent/admin/member/memberList.jsp]

1	<%@ page language="java" contentType="text/html; charset=UTF-8"
---	---

```

2   pageEncoding="UTF-8"%>
3   <%@ include file="/admin/header.jsp"%>
4   <%@ include file="/admin/sub_menu.jsp"%>
5   <script type="text/javascript">
6       function go_search()
7       {
8           document.frm.action="NonageServlet?command=admin_member_list";
9           document.frm.submit();
10      }
11  </script>
12
13  <article>
14  <h1>회원리스트</h1>
15  <form name="frm" method="post">
16  <table style="float:right; ">
17      <tr>
18          <td>
19              회원 이름
20              <input type="text" name="key">
21              <input class="btn" type="button" value="검색" onclick="go_search()">
22          </td>
23      </tr>
24  </table>
25  <br>
26  <table id="orderList">
27      <tr>
28          <th> 아디디(탈퇴여부) </th>      <th> 이름 </th>
29          <th> 이메일 </th>      <th> 우편번호 </th>
30          <th> 주소 </th> <th> 전화 </th> <th> 가입일 </th>
31      </tr>
32      <c:forEach items="${memberList}" var="memberVO">
33          <tr>
34              <td>${memberVO.id}<
35              <c:choose>
36                  <c:when test='${memberVO.useyn=="y"}'>
37                      <input type="checkbox" name="useyn" disabled="disabled">
38                  </c:when>
39                  <c:otherwise>
40                      <input type="checkbox" name="useyn"
41                      checked="checked" disabled="disabled">
42                  </c:otherwise>
43              </c:choose>
44          </td>
45          <td> ${memberVO.name} </td>
46          <td> ${memberVO.email} </td>
47          <td> ${memberVO.zipNum} </td>
48          <td> ${memberVO.address} </td>
49          <td> ${memberVO.phone} </td>

```



```

50      <td> <fmt:formatDate value="{memberVO.indate}"/></td>
51    </tr>
52  </c:forEach>
53 </table>
54 </form>
55 </article>
56 <%@ include file="/admin/footer.jsp"%>
57 </body>
58 </html>

```

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

```

.. //중복되는 내용 생략
89   } else if (command.equals("admin_member_list")) {
90     action = new AdminMemberListAction();
91   }
.. //중복되는 내용 생략

```

14-5 Q&A 게시판 관리

마지막으로 사용자가 올린 질문에 대한 답변을 달기 위해 Q&A 게시판 관리하도록 합시다. 우선 이를 위해 Q&A 게시판 정보를 저장한 테이블을 처리하는 DAO에 메소드를 추가합니다.

[직접해보세요] Q&A 게시판 테이블을 액세스하는 DAO 클래스 만들기

[파일 이름 :src\Wcom\Wnonage\Wdao\WMemberDAO.java]

```
99 //중복되는 내용 생략
100 /* *
101  * 관리자 모드에서 필요한 메소드
102  */
103 public ArrayList<QnaVO> listAllQna() {
104     ArrayList<QnaVO> qnaList = new ArrayList<QnaVO>();
105     // 게시판의 데이터를 가지고 오는 쿼리 rep:1:게시물 2: 답변
106     String sql = "select * from qna order by indate desc";
107
108     Connection conn = null;
109     PreparedStatement pstmt = null;
110     ResultSet rs = null;
111     try {
112         conn = DBManager.getConnection();
113         pstmt = conn.prepareStatement(sql);
114         rs = pstmt.executeQuery();
115         while (rs.next()) {
116             QnaVO qnaVO = new QnaVO();
117             qnaVO.setQseq(rs.getInt("qseq"));
118             qnaVO.setSubject(rs.getString("subject"));
119             qnaVO.setContent(rs.getString("content"));
120             qnaVO.setId(rs.getString("id"));
121             qnaVO.setIndate(rs.getTimestamp("indate"));
122             qnaVO.setReply(rs.getString("reply"));
123             qnaVO.setRep(rs.getString("rep"));
124             qnaList.add(qnaVO);
125         }
126     } catch (Exception e) {
127         e.printStackTrace();
128     }
129     return qnaList;
130 }
131
132 public void updateQna(QnaVO qnaVO) {
133     String sql = "update qna set reply=?, rep='2' where qseq=?";
134
135     Connection conn = null;
136     PreparedStatement pstmt = null;
137     try {
```

```

138     conn = DBManager.getConnection();
139     pstmt = conn.prepareStatement(sql);
140     pstmt.setString(1, qnaVO.getReply());
141     pstmt.setInt(2, qnaVO.getQseq());
142     pstmt.executeUpdate();
143 } catch (Exception e) {
144     e.printStackTrace();
145 } finally {
146     DBManager.close(conn, pstmt);
147 }
148 }
149 }

```

화면 좌측의 Q&A 게시판 리스트를 클릭하면 “NonageServlet?command=admin_qna_list” 요청이 발생합니다. 이를 위한 요청 클래스를 만듭시다.

[직접해보세요] Q&A 게시판 리스트를 출력하기 위한 액션 클래스[파일이름 : srcWcomWnonageWadminWcontrollerWactionWAdminQnaListAction .java]

```

1 package com.nonage.admin.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
10 import com.nonage.controller.action.Action;
11 import com.nonage.dao.QnaDAO;
12 import com.nonage.dto.QnaVO;
13
14 public class AdminQnaListAction implements Action {
15
16     @Override
17     public void execute(HttpServletRequest request, HttpServletResponse response)
18         throws ServletException, IOException {
19
20         String url = "admin/qna/qnaList.jsp";
21
22         QnaDAO qnaDAO = QnaDAO.getInstance();
23         ArrayList<QnaVO> qnaList = qnaDAO.listAllQna();
24
25         request.setAttribute("qnaList", qnaList);
26
27         request.getRequestDispatcher(url).forward(request, response);
28     }
29 }

```

[직접해보세요] Q&A 게시판 리스트를 출력하는 JSP 파일

[파일이름 : WebContent/admin/qna/qnaList.jsp]

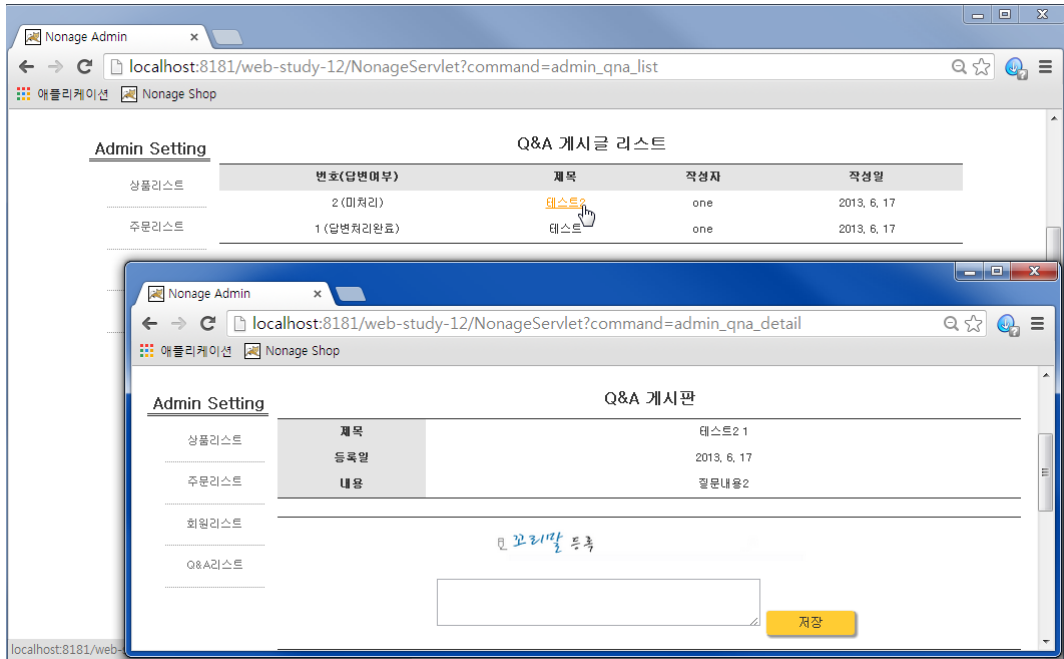
```
1  <%@ page language="java" contentType="text/html; charset=UTF-8"
2    pageEncoding="UTF-8"%>
3  <%@ include file="/admin/header.jsp"%>
4  <%@ include file="/admin/sub_menu.jsp"%>
5  <script type="text/javascript">
6    function go_view(qseq) {
7      var theForm = document.frm;
8      theForm.qseq.value = qseq;
9      theForm.action = "NonageServlet?command=admin_qna_detail";
10     theForm.submit();
11   }
12 </script>
13
14 <article>
15 <h1>qna 리스트</h1>
16 <form name="frm" method="post">
17 <input type="hidden" name="qseq">
18 <table id="orderList">
19   <tr>
20     <th>번호(답변여부)</th> <th>제목</th> <th>작성자</th> <th>작성일</th>
21   </tr>
22   <c:forEach items="${qnaList}" var="qnaVO">
23     <tr>
24       <td>
25         ${qnaVO.qseq}
26         <c:choose>
27           <c:when test='${qnaVO.rep=="1"}>(미처리)</c:when>
28           <c:otherwise>(답변처리완료)</c:otherwise>
29         </c:choose>
30       </td>
31       <td>
32         <a href="#" onClick="javascript:go_view('${qnaVO.qseq}')">
33           ${qnaVO.subject}
34         </a>
35       </td>
36       <td> ${qnaVO.id} </td>
37       <td> <fmt:formatDate value="${qnaVO.indate}"/></td>
38     </tr>
39   </c:forEach>
40 </table>
41 </form>
42 </article>
43 <%@ include file="/admin/footer.jsp"%>
44 </body>
45 </html>
```

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

```

.. //중복되는 내용 생략
91     } else if (command.equals("admin_qna_list")) {
92         action = new AdminQnaListAction();
93     }
.. //중복되는 내용 생략
  
```

게시글 리스트에서 답변 처리가 되지 않은 게시글에 답변을 달기 위해서는 제목을 클릭합니다.



14-015

클릭한 게시글에 대한 답변을 달 수 있는 게시글 상세 보기 화면으로 이동하도록
 “NonageServlet?command=qna_list_detail” 요청이 발생합니다. 이 요청을 처리할 액션 클래스를 만듭니다.

[직접해보세요] 게시글 상세 보기를 위한 액션 클래스[파일이름 :
 srcWcomWnonageWadminWcontrollerWactionWAdminQnaDetailAction.java]

```

1 package com.nonage.admin.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.QnaDAO;
10 import com.nonage.controller.action.Action;
11 import com.nonage.dto.QnaVO;
12
13 public class AdminQnaDetailAction implements Action {
14
15     @Override
16     public void execute(HttpServletRequest request, HttpServletResponse response)
17         throws ServletException, IOException {
18
19         String url = "admin/qna/qnaDetail.jsp";
20
21         String qseq = request.getParameter("qseq").trim();
22
23         QnaDAO qnaDAO = QnaDAO.getInstance();
24         QnaVO qnaVO = qnaDAO.getQna(Integer.parseInt(qseq));
25
26         request.setAttribute("qnaVO", qnaVO);
27
28         request.getRequestDispatcher(url).forward(request, response);
29     }
30 }

```

[직접해보세요] 질문에 대한 답변을 입력할 수 있는 JSP 페이지

[파일이름 : WebContent/qna/qnaDetail.jsp]

```

1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <%@ include file="/admin/header.jsp"%>
4 <%@ include file="/admin/sub_menu.jsp"%>
5 <script type="text/javascript">
6   function go_list(){
7       var theForm = document.frm;
8       theForm.action="NonageServlet?command=admin_qna_list";
9       theForm.submit();
10  }
11
12  function go_rep(qseq){
13      var theForm = document.frm;
14      theForm.qseq.value=qseq;
15      theForm.action="NonageServlet?command=admin_qna_repsave";
16      theForm.submit();
17  }
18 </script>
19 <article>
20 <h1>Q&A 게시판</h1>
21 <form name="frm" method="post">

```

```

22 <input type="hidden" name="qseq">
23 <table id="orderList">
24 <tr>
25 <th width="20%">제목</th>
26 <td> ${qnaVO.subject} ${qnaVO.rep} </td>
27 </tr>
28 <tr>
29 <th>등록일</th>
30 <td> <fmt:formatDate value="${qnaVO.indate}"/> </td>
31 </tr>
32 <tr>
33 <th>내용</th>
34 <td> ${qnaVO.content}</td>
35 </tr>
36 </table>
37 <c:choose>
38 <c:when test='${qnaVO.rep=="1"}'>
39 <table id="orderList">
40 <tr>
41 <td colspan="2">
42 
43 </td>
44 </tr>
45 <tr>
46 <td colspan="2">
47 <textarea name="reply" rows="3" cols="50"></textarea>
48 <input type="button" class="btn" value="저장"
49 onClick="go_rep('${qnaVO.qseq}')">
50 </td>
51 </tr>
52 </table>
53 <br>
54 </c:when>
55 <c:otherwise>
56 <table id="orderList">
57 <tr>
58 <th>댓글</th>
59 <td> ${qnaVO.reply}</td>
60 </tr>
61 </table>
62 </c:otherwise>
63 </c:choose>
64 <input type="button" class="btn" value="목록" onClick="go_list()">
65 </form>
66 </article>
67 <%@ include file="/admin/footer.jsp"%>
68 </body>
69 </html>

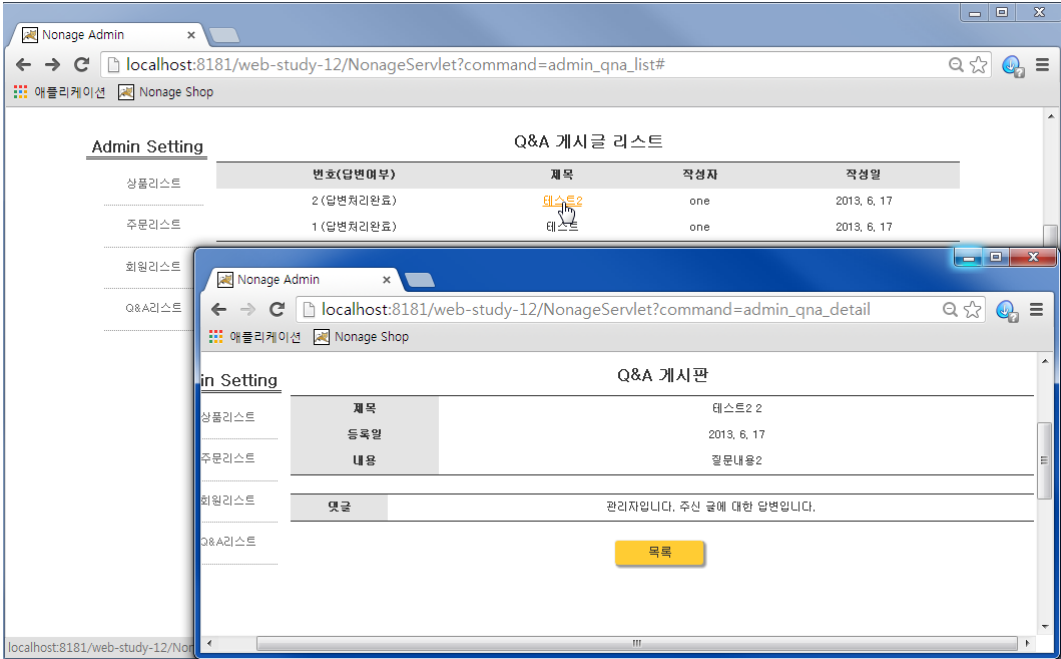
```

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

```

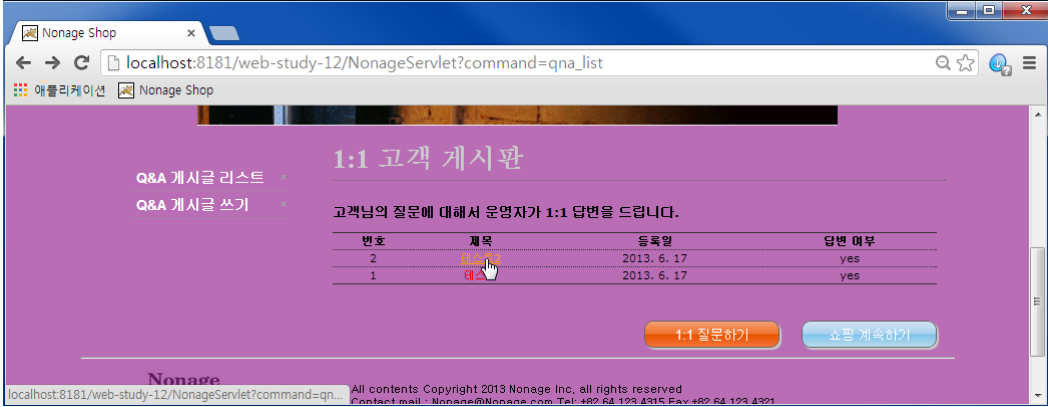
.. //중복되는 내용 생략
93     } else if (command.equals("admin_qna_detail")) {
94         action = new AdminQnaDetailAction();
95     }
.. //중복되는 내용 생략
  
```

답변을 달고 [저장] 버튼을 클릭하면 게시글 리스트로 이동합니다. 답변을 단 게시글에 대
 해서 번호 앞에 답변 처리 완료되었다는 메시지로 바뀐 것을 확인할 수 있습니다.

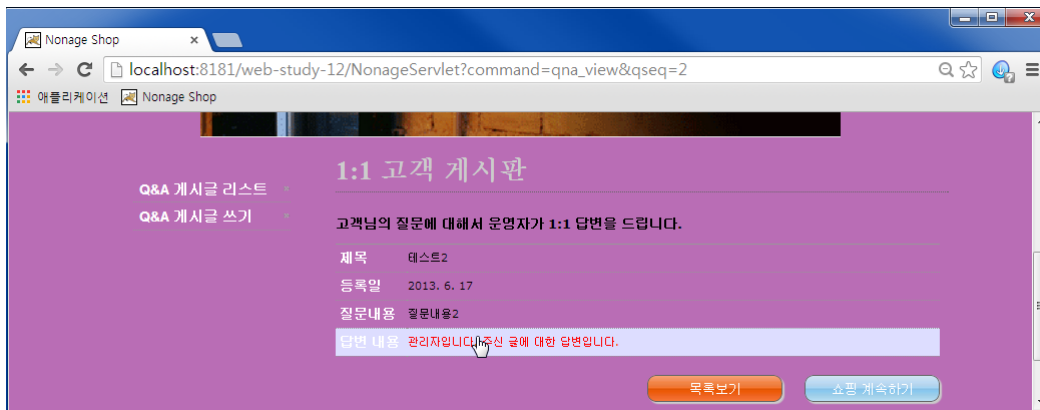


14-017

제목을 클릭하면 게시글 상세 보기 페이지에서도 답변이 달린 것을 확인할 수 있습니다.
 관리자 페이지뿐만 아니라 사용자 페이지에서도 관리자가 단 답변을 확인할 수 있습니다.



14-019



14-018

답변을 단 후에 [저장] 버튼을 클릭하면 "NonageServlet?command=admin_qna_repsave" 이 요청되어 입력한 답변 내용으로 게시글을 갱신합니다. 이를 처리하기 위한 액션 클래스를 만듭시다.

[직접해보세요] 게시글에 답변 달기 위한액션 클래스[파일이름 :
srcWcomWnonageWadminWcontrollerWactionWAdminQnaResaveAction.java]

```

1 package com.nonage.admin.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.QnaDAO;
10 import com.nonage.controller.action.Action;
11 import com.nonage.dto.QnaVO;
12
13 public class AdminQnaResaveAction implements Action {
14
15     @Override
16     public void execute(HttpServletRequest request, HttpServletResponse response)
17         throws ServletException, IOException {
18
19         String url = "NonageServlet?command=admin_product_list";
20
21         String qseq = request.getParameter("qseq").trim();
22         String reply = request.getParameter("reply").trim();
23
24         QnaVO qnaVO = new QnaVO();
25         qnaVO.setQseq(Integer.parseInt(qseq));
26         qnaVO.setReply(reply);
27

```

```

28     QnaDAO qnaDAO = QnaDAO.getInstance();
29     qnaDAO.updateQna(qnaVO);
30
31     response.sendRedirect(uri);
32 }
33 }

```

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

```

.. //중복되는 내용 생략
95     } else if (command.equals("admin_qna_repsave")) {
96         action = new AdminQnaResaveAction();
97     }
.. //중복되는 내용 생략

```

마무리 하며...

지금까지 온라인 쇼핑몰을 구축해보았습니다. <백견불여일타 JSP&Servlet>의 본문에서 언급하지 않은 기술들도 나와 당황하였을 수도 있습니다. 하지만, 본문을 충실히 이해하고 API 문서 등을 잘 참조하며 따라했다면 더 큰 성장을 이루었을 거라 확신합니다. 이 특별부록은 한번 따라해보고 끝내지 말고 다양한 수정버전을 만들어보기를 추천합니다. 레이아웃도 바꾸어보고 기능이나 데이터베이스도 추가해보기 바랍니다. 그러다보면 부가적인 클라이언트 기술이나 DB 등의 이해 폭도 훨씬 넓어질 것입니다.

여러분의 앞날에 행복이 항상 함께하기를 바라겠습니다.

감사합니다.

저자 성윤정 (2014년 7월 17일)