

30장 스프링으로 답변형 게시판 만들기

30.1 기존 소스 코드 변경하기

30.9 새 글 추가 시 여러 이미지 파일 첨부하기

30.2 마이바티스 관련 XML 설정하기

30.10 글상세창에 여러 이미지 표시하기

30.3 타일즈 설정하기

30.4 게시판 목록 표시하기

30.5 새 글 추가하기

30.6 글상세창 구현하기

30.7 글 수정하기

30.8 글 삭제하기

13.1 기존 소스 코드 변경하기

1. pro30의 pom.xml에 27장의 pom.xml의 설정을 복사하여 타일즈와 마이바티스 관련 라이브러리를 추가합니다

코드 30-1 pro30/pom.xml

```
...
<!-- 마이바티스 -->
<dependency>
  <groupId>commons-beanutils</groupId>
  <artifactId>commons-beanutils</artifactId>
  <version>1.8.0</version>
</dependency>
...
<!-- 타일즈 관련 라이브러리 -->
<dependency>
  <groupId>org.apache.tiles</groupId>
  <artifactId>tiles-core</artifactId>
  <version>2.2.2</version>
</dependency>
...
```

13.1 기존 소스 코드 변경하기

2. lib 폴더를 만든 후 오라클 드라이버를 복사하여 붙여 넣습니다(27.7.1절 참조).



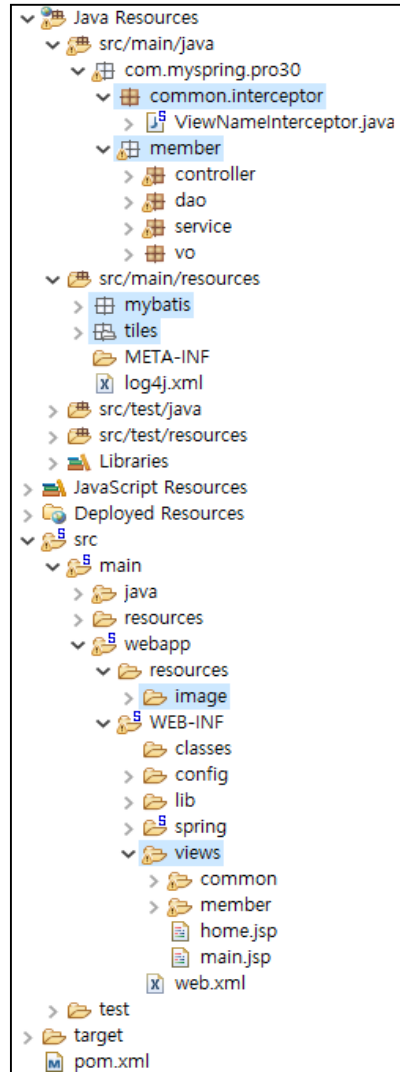
13.1 기존 소스 코드 변경하기

3. 인터셉터를 추가하겠습니다. com/myspring/pro30 패키지 하위에 다시 common/interceptor 패키지를 만든 후 27장의 인터셉터 실습 파일을 그대로 복사하여 붙여 넣습니다



13.1 기존 소스 코드 변경하기

4. 27장의 회원 관리 기능에 사용된 XML 설정 파일, 자바 클래스, JSP 파일 순으로 동일한 패키지과 폴더를 만든 후 복사하여 붙여 넣습니다.



13.1 기존 소스 코드 변경하기

5. 회원 기능 중 게시판 실습에 필요한 자바 코드와 JSP 파일을 수정해 보겠습니다. 컨트롤러에 /main.do로 요청하면 main.jsp를 표시해 주는 main() 메서드를 추가합니다.

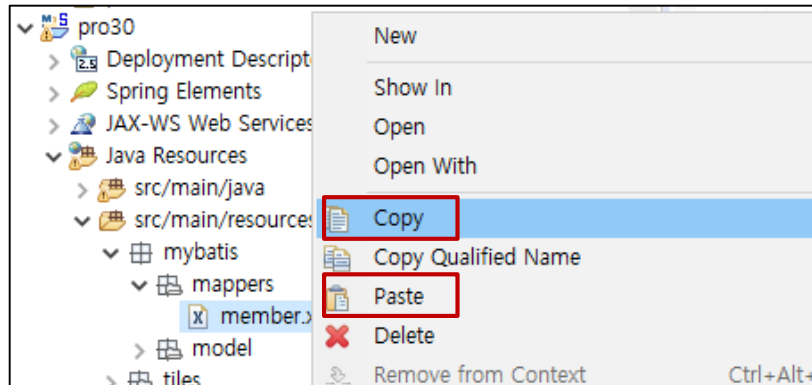
코드 30-2 pro30/src/main/java/com/myspring/pro30/member/controller/MemberControllerImpl.java

```
package com.myspring.pro30.member.controller;
.....
@Controller("memberController")
public class MemberControllerImpl implements MemberController {
    @Autowired
    private MemberService memberService;
    @Autowired
    MemberVO memberVO ;
    @RequestMapping(value = { "/", "/main.do"}, method = RequestMethod.GET)
    private ModelAndView main(HttpServletRequest request,
                            HttpServletResponse response) throws Exception {
        String viewName = (String)request.getAttribute("viewName");
        ModelAndView mav = new ModelAndView();
        mav.setViewName(viewName);
        return mav;
    }
    ...
}
```

/pro30/main.do로 요청 시
메인 페이지를 보여 줍니다.

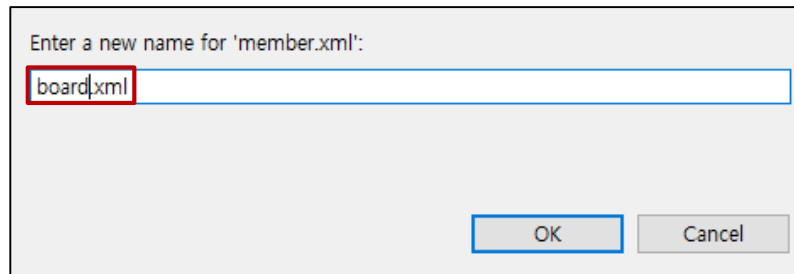
30.2 마이바티스 관련 XML 설정하기

1. src/main/resources/ 패키지에 게시판 기능 관련 매퍼 파일인 board.xml을 추가합니다.
2. member.xml 위에서 마우스 오른쪽 버튼을 클릭한 후 Copy를 선택하고 다시 Paste를 선택합니다.

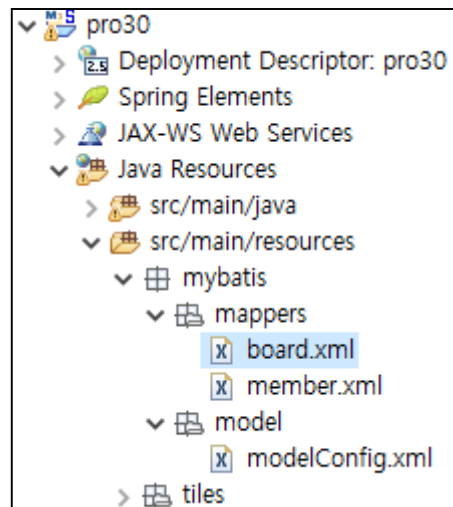


30.2 마이바티스 관련 XML 설정하기

3. 복사한 member.xml의 파일 이름을 board.xml로 변경한 후 OK를 클릭합니다.



4. 게시판 관련 매퍼 파일인 board.xml이 생성된 것을 확인할 수 있습니다.



30.2 마이바티스 관련 XML 설정하기

5. 기존 파일 내용을 삭제한 후 <select> 태그로 id가 selectAllArticlesList인 SQL문을 추가합니다.

코드 30-3 pro30/src/main/resources/mybatis/mappers/board.xml

```
...
<mapper namespace="mapper.board">
  <resultMap id="articlesResult" type="articleVO">
    <result property="level" column="level" />
    <result property="articleNO" column="articleNO" />
    <result property="parentNO" column="parentNO" />
    <result property="title" column="title" />
    <result property="content" column="content" />
    <result property="writeDate" column="writeDate" />
    <result property="imageFileName" column="imageFileName" />
  </resultMap>

  <select id="selectAllArticlesList" resultMap="articlesResult" >
    <![CDATA[
      SELECT level, articleNO, parentNO, title, content, writeDate, imageFileName, id
      from t_board
      START WITH parentNO=0
      CONNECT BY PRIOR articleNO=parentNO
      ORDER SIBLINGS BY articleNO DESC
    ]]>
  </select>
...
```

네임스페이스를 변경합니다.

글 정보를 저장할 resultMap을 설정합니다.

모든 글을 조회하는 SQL문을 작성합니다.

30.2 마이바티스 관련 XML 설정하기

6. modelConfig.xml에는 매퍼 파일에서 사용할 articleVO를 추가합니다.

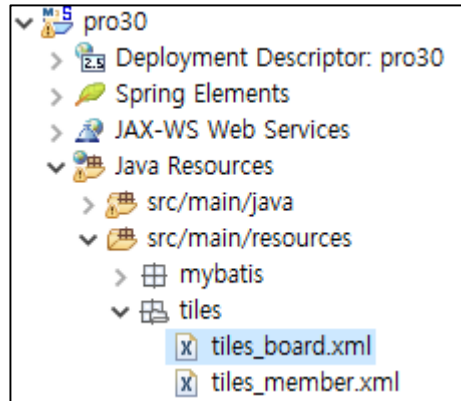
코드 30-4 pro30/src/main/resources/mybatis/model/modelConfig.xml

```
...  
<configuration>  
  <typeAliases>  
    <typeAlias type="com.myspring.pro30.member.vo.MemberVO" alias="memberVO" />  
    <typeAlias type="com.myspring.pro30.board.vo.ArticleVO" alias="articleVO" />  
  </typeAliases>  
</configuration>
```

매퍼 파일에서 사용할 articleVO를 설정합니다.

30.3 타일즈 설정하기

1. 30.2절에서와 마찬가지로 tiles_member.xml을 복사해 붙여 넣은 후 파일 이름을 tiles_board.xml로 변경합니다.



30.3 타일즈 설정하기

2. 기존 내용을 삭제한 후 게시판 화면 기능은 tiles_board.xml에 설정해서 사용합니다. 글목록창을 <definition> 태그로 설정합니다.

코드 30-5 pro30/src/main/resources/tiles/tiles_board.xml

```
...  
<tiles-definitions>  
  <definition name="/board/listArticles" extends="baseLayout">  
    <put-attribute name="title" value="글목록창" />  
    <put-attribute name="body" value="/WEB-INF/views/board/listArticles.jsp" />  
  </definition>  
</tiles-definitions>
```

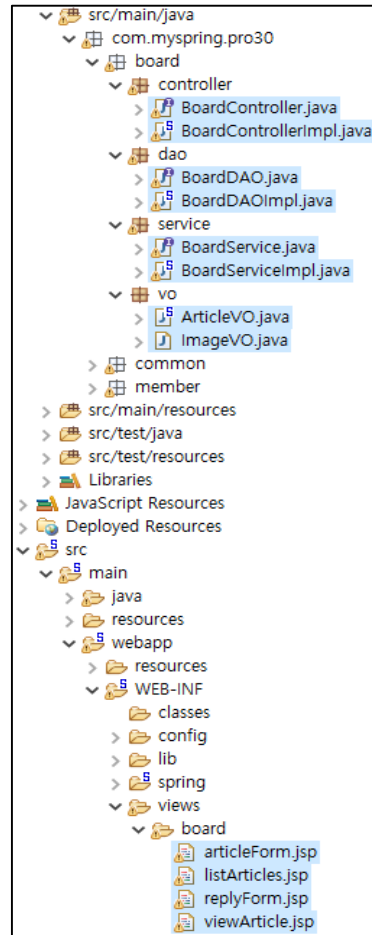
/board/listArticles 요청에 대해 같은 경로의 listArticles.jsp를 표시합니다.

tiles_member.xml의 기본 레이아웃을 상속받습니다.

30.4 게시판 목록 표시하기

• 30.4.1 자바 클래스 구현하기

1. 먼저 src/main/java 패키지 하위에 board 패키지를 만든 후 그림 30-8처럼 패키지를 구성합니다.
그리고 JSP는 17장에서 실습한 게시판 관련 JSP 파일을 재사용할 것이므로 프로젝트 pro17의 board07 폴더에 있는 JSP 파일을 모두 복사해 views/board 폴더에 붙여 넣습니다



30.4 게시판 목록 표시하기

2. 컨트롤러 클래스인 BoardControllerImpl를 다음과 같이 작성합니다. 브라우저에서 요청하면 모든 글 정보를 조회한 후 ModelAndView 객체에 바인딩하여 JSP로 전달합니다.

코드 30-6 pro30/src/main/java/com/myspring/pro30/board/controller/BoardControllerImpl.java

```
package com.myspring.pro30.board.controller;

...

@Controller("boardController")
public class BoardControllerImpl implements BoardController{
    private static final String CURR_IMAGE_REPO_PATH = "C:\\board\\article_imagefile";
    @Autowired
    BoardService boardService;
    @Autowired
    ArticleVO articleVO;

    @Override
    @RequestMapping(value= "/board/listArticles.do",
                    method = {RequestMethod.GET, RequestMethod.POST})
    public ModelAndView listArticles(HttpServletRequest request,
                                    HttpServletResponse response) throws Exception {
    {
        String viewName = (String)request.getAttribute("viewName");
        List articlesList = boardService.listArticles();
        ModelAndView mav = new ModelAndView(viewName);
        mav.addObject("articlesList", articlesList);
        return mav;
    }
}
```

인터셉터에서 전달된 뷰이름을 가져옵니다.

모든 글 정보를 조회합니다.

조회한 글 정보를 바인딩한 후 JSP로 전달합니다.

30.4 게시판 목록 표시하기

3. 서비스 클래스에서는 boardDAO의 selectAllArticlesList() 메서드를 호출합니다.

코드 30-7 pro30/src/main/java/com/myspring/pro30/board/service/BoardServiceImpl.java

```
package com.myspring.pro30.board.service;

...

@Service("boardService")
@Transactional(propagation = Propagation.REQUIRED)
public class BoardServiceImpl implements BoardService{
    @Autowired
    BoardDAO boardDAO;

    public List<ArticleVO> listArticles() throws Exception{
        List<ArticleVO> articlesList = boardDAO.selectAllArticlesList();
        return articlesList;
    }
}
```

boardDAO의 selectAllArticlesList() 메서드를 호출합니다.

30.4 게시판 목록 표시하기

4. DAO 클래스는 다시 매퍼 파일인 board.xml에서 지정한 id로 SQL문을 실행합니다.

코드 30-8 pro30/src/main/java/com/myspring/pro30/board/dao/BoardDAOImpl.java

```
package com.myspring.pro30.board.dao;

...
@Repository("boardDAO")
public class BoardDAOImpl implements BoardDAO {
    @Autowired
    private SqlSession sqlSession;

    @Override
    public List selectAllArticlesList() throws Exception {
        List<ArticleVO> articlesList =
            sqlSession.selectList("mapper.board.selectAllArticlesList");
        return articlesList;
    }
}
```

id가 selectAllArticlesList인 SQL문을 요청합니다.

30.4 게시판 목록 표시하기

5. 글 정보를 저장할 ArticleVO 클래스를 다음과 같이 작성합니다

코드 30-9 pro30/src/main/java/com/myspring/pro30/board/vo/ArticleVO.java

```
package com.myspring.pro30.board.vo;

...

@Component("articleVO")
public class ArticleVO {
    private int level;
    private int articleNO;
    private int parentNO;
    private String title;
    private String content;
    private String imageFileName;
    private String id;
    private Date writeDate;

    //각 속성에 대한 getter/setter
    ...
}
```

30.4 게시판 목록 표시하기

• 30.4.2 JSP 구현하기

1. 글목록창에 해당하는 listArticles.jsp를 다음과 같이 작성합니다.

코드 30-10 pro30/src/main/webapp/WEB-INF/views/board/listArticles.jsp

```
...
<c:when test="${articlesList != null }" >
  <c:forEach var="article" items="${articlesList }" varStatus="articleNum" >
    <tr align="center">
      <td width="5%">${articleNum.count}</td>
      <td width="10%">${article.id }</td>
      <td align='left' width="35%">
        <span style="padding-right:30px"></span>
        <c:choose>
          <c:when test='${article.level > 1 }'>
            <c:forEach begin="1" end="${article.level }" step="1">
              <span style="padding-left:20px"></span>
            </c:forEach>
            <span style="font-size:12px;">[답변]</span>
            <a class='cls1' href="${contextPath}/board/viewArticle.do?
              articleNO=${article.articleNO}">${article.title}</a>
          </c:when>
          <c:otherwise>
            <a class='cls1' href="${contextPath}/board/viewArticle.do?
              articleNO=${article.articleNO}">${article.title }</a>
          </c:otherwise>
        </c:choose>
      </td>
      <td width="10%">${article.writeDate}</td>
    </tr>
  </c:forEach>
</c:when>
</c:choose>
...
```

level이 1보다 크면 답글을 표시합니다.

level이 1보다 작으면 부모 글을 표시합니다.

30.4 게시판 목록 표시하기

2. 왼쪽 메뉴의 각 항목을 클릭할 경우 해당 목록창으로 이동하도록 작성합니다

코드 30-11 pro30/src/main/webapp/WEB-INF/views/common/side.jsp

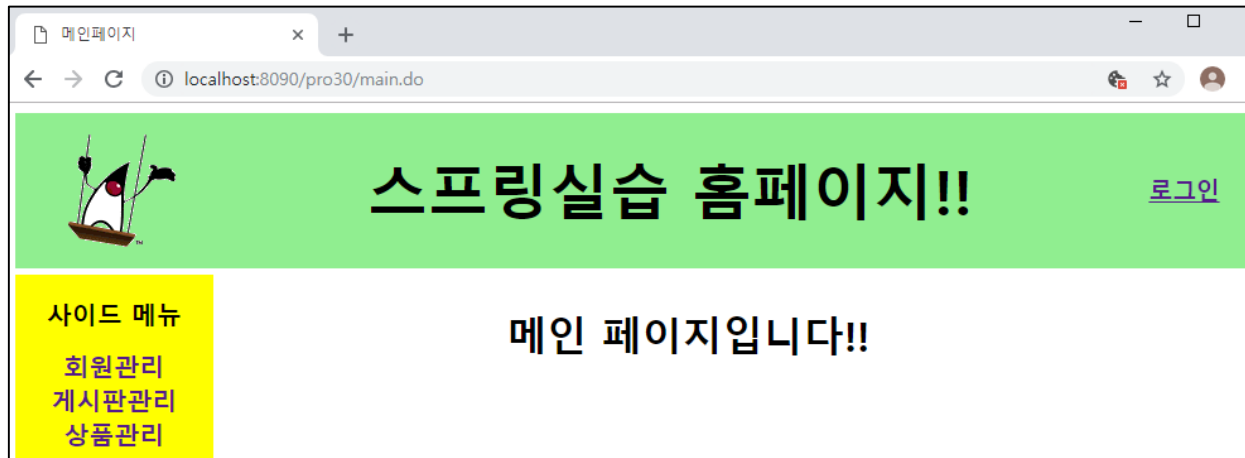
```
...  
<h1>  
  <a href="${contextPath}/member/listMembers.do" class="no-underline">회원관리</a><br>  
  <a href='${contextPath}/board/listArticles.do' class="no-underline">게시판관리</a><br>  
  <a href="#" class="no-underline">상품관리</a><br>  
</h1>  
...
```

회원관리 클릭 시 회원 목록창으로 이동합니다.

게시판관리 클릭 시 글목록창으로 이동합니다.

30.4 게시판 목록 표시하기

3. <http://localhost:8090/pro30/main.do>로 요청하면 메인 페이지가 나타납니다



30.4 게시판 목록 표시하기

4. 왼쪽 메뉴에 있는 게시판관리를 클릭하면 글목록창이 나타납니다



스프링실습 홈페이지!!

[로그인](#)

사이드 메뉴

[회원관리](#)

[게시판관리](#)

[상품관리](#)

글번호	작성자	제목	작성일
1	hong	상품평입니다.	2018-09-18
2	lee	[답변] 이용 후기입니다.	2018-11-22
3	kim	김유신입니다.	2018-09-18
4	lee	[답변] 이용 후기입니다.	2018-09-18
5	hong	안녕하세요	2018-09-18
6	lee	[답변] 상품후기입니다..	2018-09-18
7	hong	[답변] 답변입니다.	2018-09-18
8	lee	[답변] 답변입니다.	2018-09-18
9	hong	테스트글입니다.	2018-09-18

글쓰기



30.5 새 글 추가하기

- 30.5.1 XML 파일 설정하기

1. 파일 업로드와 관련된 설정을 실습해 보겠습니다. pom.xml에 파일 첨부 기능과 관련된 라이브러리를 설정합니다.

코드 30-12 pro30/pom.xml

```
...  
<dependency>  
  <groupId>commons-fileupload</groupId>  
  <artifactId>commons-fileupload</artifactId>  
  <version>1.2.1</version>  
</dependency>  
  
<dependency>  
  <groupId>commons-io</groupId>  
  <artifactId>commons-io</artifactId>  
  <version>1.4</version>  
</dependency>  
...
```

30.5 새 글 추가하기

2. servlet-context.xml에 다중 파일 업로드 기능을 위한 멀티파트 리졸버 빈을 설정합니다.

코드 30-13 pro30/src/main/webapp/WEB-INF/spring/appServlet/servlet-context.xml

```
...  
    <beans:bean id="multipartResolver"  
        class="org.springframework.web.multipart.commons.  
            CommonsMultipartResolver">  
        <beans:property name="maxUploadSize" value="52428800" />  
        <beans:property name="maxInMemorySize" value="1000000" />  
        <beans:property name="defaultEncoding" value="utf-8" />  
    </beans:bean>  
...
```

30.5 새 글 추가하기

3. 매퍼 파일인 board.xml에 새 글 추가 기능에 사용할 insert문을 작성합니다.

코드 30-14 pro30/src/main/resources/mybatis/mappers/board.xml

```
...
<insert id="insertNewArticle" parameterType="java.util.Map">
  <![CDATA[
    INSERT into t_board(articleNO, title, content, imageFileName, id)
    VALUES("#{articleNO}",#{title}, #{content}, #{imageFileName}, #{id})
  ]]>
</insert>

<select id="selectNewArticleNO" resultType="int" >
  <![CDATA[
    SELECT max(articleNO)+1 from t_board
  ]]>
</select>
...
```

글 정보를 Map으로 전달합니다.

추가하는 새 글에 대한 글 번호를 가져옵니다.

30.5 새 글 추가하기

4. 글쓰기창을 타일즈 파일인 tiles_board.xml에 설정합니다.

코드 30-15 pro30/src/main/resources/tiles/tiles_board.xml

...

```
<definition name="/board/articleForm" extends="baseLayout">
  <put-attribute name="title" value="글쓰기창" />
  <put-attribute name="body" value="/WEB-INF/views/board/articleForm.jsp" />
</definition>
```

...

/board/articleForm 요청에 대해 글쓰기창을 나타냅니다.

30.5 새 글 추가하기

- 30.5.2 자바 클래스 구현하기

글쓰기 시 로그인 과정

- ① 글목록창(listArticles.jsp) 요청 시 미리 세션의 isLogOn 속성을 자바스크립트 함수의 인자로 저장합니다.
- ② 글쓰기를 클릭하면 자바스크립트 함수에서 isLogOn 속성 값을 체크하여 true가 아니면 memberController로 로그인창을 요청하면서 다음에 수행할 URL을 action 값으로 전송합니다.
- ③ memberController는 action 값을 세션에 저장합니다.
- ④ 로그인창에서 ID와 비밀번호를 입력하여 memberController로 전송한 후 로그인에 성공하면 세션의 action 속성 값을 가져와서 글쓰기창으로 바로 이동합니다.
- ⑤ ❷에서 isLogOn 속성이 true이면 바로 글쓰기창으로 이동합니다.

30.5 새 글 추가하기

1. MemberControllerImpl 클래스를 다음과 같이 작성합니다.

코드 30-16 pro30/src/main/java/com/myspring/pro30/member/controller/MemberControllerImpl.java

```
package com.myspring.pro30.member.controller;
...
@Controller("memberController")
public class MemberControllerImpl implements MemberController {
    @Autowired
    private MemberService memberService;
    @Autowired
    MemberVO memberVO ;
    ...
    @Override
    @RequestMapping(value = "/member/login.do", method = RequestMethod.POST)
    public ModelAndView login(@ModelAttribute("member") MemberVO member,
                             RedirectAttributes rAttr,
                             HttpServletRequest request,
                             HttpServletResponse response) throws Exception {
        ModelAndView mav = new ModelAndView();
        memberVO = memberService.login(member);
        if(memberVO != null) {
            HttpSession session = request.getSession();
            session.setAttribute("member", memberVO);
            session.setAttribute("isLogOn", true);
        }
    }
}
```

리다이렉트 시 매개변수를 전달합니다.

로그인 성공 시 조건문을 수행합니다.

30.5 새 글 추가하기

```
String action = (String)session.getAttribute("action");
session.removeAttribute("action");
if(action != null) {
    mav.setViewName("redirect:"+action);
}else {
    mav.setViewName("redirect:/member/listMembers.do");
}
}else {
    rAttr.addAttribute("result","loginFailed");
    mav.setViewName("redirect:/member/loginForm.do");
}
return mav;
}
```

로그인 성공 시 세션에 저장된 action 값을 가져옵니다.

action 값이 null이 아니면 action 값을 뷰이름으로 지정해 글쓰기창으로 이동합니다.

로그인 실패 시 다시 로그인창으로 이동합니다.

```
@Override
@RequestMapping(value = "/member/logout.do", method = RequestMethod.GET)
public ModelAndView logout(HttpServletRequest request,
                           HttpServletResponse response) throws Exception {
    HttpSession session = request.getSession();
    session.removeAttribute("member");
    session.removeAttribute("isLogOn");
    ModelAndView mav = new ModelAndView();
    mav.setViewName("redirect:/member/listMembers.do");
    return mav;
}
```

30.5 새 글 추가하기

p.1084 login() 메서드 수정

로그인 후 수행할 글쓰기 요청명을 action에 저장합니다.
로그인 성공 후 바로 글쓰기창으로 이동합니다.

```
@RequestMapping(value = "/member/*Form.do", method = RequestMethod.GET)
private ModelAndView form(@RequestParam(value= "result", required=false) String result,
                          @RequestParam(value= "action", required=false) String action,
                          HttpServletRequest request,
                          HttpServletResponse response) throws Exception {
    String viewName = (String)request.getAttribute("viewName");
    HttpSession session = request.getSession();
    session.setAttribute("action", action);
    ModelAndView mav = new ModelAndView();
    mav.addObject("result", result);
    mav.setViewName(viewName);
    return mav;
}
...
}
```

글쓰기창 요청명을 action 속성으로 세션에 저장합니다.

30.5 새 글 추가하기

2. 이번에는 글쓰기 기능을 구현할 BoardControllerImpl 클래스를 작성합니다.

코드 30-17 pro30/src/main/java/com/myspring/pro30/board/controller/BoardControllerImpl.java

```
package com.myspring.pro30.board.controller;

...
@Controller("boardController")
public class BoardControllerImpl implements BoardController{
    private static final String ARTICLE_IMAGE_REPO = "C:\\board\\article_image";
    @Autowired
    BoardService boardService;
    @Autowired
    ArticleVO articleVO;
    ...
    @Override
    @RequestMapping(value="/board/addNewArticle.do", method = RequestMethod.POST)
    @ResponseBody
    public ResponseEntity addNewArticle(MultipartHttpServletRequest multipartRequest,
                                         HttpServletResponse response) throws Exception {
        multipartRequest.setCharacterEncoding("utf-8");
        Map<String, Object> articleMap = new HashMap<String, Object>();
        Enumeration enu=multipartRequest.getParameterNames();
        while(enu.hasMoreElements()){
            String name=(String)enu.nextElement();
            String value=multipartRequest.getParameter(name);
            articleMap.put(name,value);
        }
    }
}
```

글 정보를 저장하기 위한
articleMap을 생성합니다.

글쓰기창에서 전송된 글 정보를
Map에 key/value로 저장합니다.

30.5 새 글 추가하기

```
String imageFileName= upload(multipartRequest);
HttpSession session = multipartRequest.getSession();
MemberVO memberVO = (MemberVO) session.getAttribute("member");
String id = memberVO.getId();
articleMap.put("parentNO", 0);
articleMap.put("id", id);
articleMap.put("imageFileName", imageFileName);

String message;
ResponseEntity resEnt=null;
HttpHeaders responseHeaders = new HttpHeaders();
responseHeaders.add("Content-Type", "text/html; charset=utf-8");
```

업로드한 이미지 파일 이름을 가져옵니다.

세션에 저장된 회원 정보로부터 회원 ID를 가져옵니다.

회원 ID, 이미지 파일 이름, 부모 글 번호를 articleMap에 저장합니다.

30.5 새 글 추가하기

```

try {
    // 글 정보가 저장된 articleMap을 Service 클래스의 addArticle() 메서드로 전달합니다.
    int articleNO = boardService.addNewArticle(articleMap);
    if(imageFileName!=null && imageFileName.length()!=0) {
        File srcFile = new
            File(ARTICLE_IMAGE_REPO+"\\temp"+"\\ "+imageFileName);
        File destDir = new File(ARTICLE_IMAGE_REPO+"\\ "+articleNO);
        FileUtils.moveFileToDirectory(srcFile, destDir,true);
    }
    message = "<script>";
    // 글 정보를 추가한 후 업로드한 이미지 파일을
    // 글 번호로 명명한 폴더로 이동합니다.
    message += " alert('새글을 추가했습니다.');"
    message += " location.href='"
        +multipartRequest.getContextPath()+"/board/listArticles.do'; ";
    message += "</script>";
    resEnt = new ResponseEntity(message, responseHeaders, HttpStatus.CREATED);
} catch (Exception e) {
    // 새 글을 추가한 후 메시지를 전달합니다.
    File srcFile = new File(ARTICLE_IMAGE_REPO+"\\temp"+"\\ "+imageFileName);
    srcFile.delete();
    // 오류 발생 시 오류 메시지를 전달합니다.
    message = " <script>";
    message += " alert('오류가 발생했습니다. 다시 시도해 주세요.');"
    message += " location.href='"
        + multipartRequest.getContextPath()+"/board/articleForm.do'; ";
    message += "</script>";
    resEnt = new ResponseEntity(message, responseHeaders, HttpStatus.CREATED);
    e.printStackTrace();
}
return resEnt;
}

```


30.5 새 글 추가하기

글쓰기창을 나타냅니다.

```
@RequestMapping(value = "/board/*Form.do", method = RequestMethod.GET)
private ModelAndView form(HttpServletRequest request, HttpServletResponse response)
throws Exception {
    String viewName = (String)request.getAttribute("viewName");
    ModelAndView mav = new ModelAndView();
    mav.setViewName(viewName);
    return mav;
}
```

```
private String upload(MultipartHttpServletRequest multipartRequest)
throws Exception{
    String imageFileName= null;
    Map<String, String> articleMap = new HashMap<String, String>();
    Iterator<String> fileNames = multipartRequest.getFileNames();
    while(fileNames.hasNext()){
        String fileName = fileNames.next();
        MultipartFile mFile = multipartRequest.getFile(fileName);
        imageFileName=mFile.getOriginalFilename();
        File file = new File(ARTICLE_IMAGE_REPO + "\\" + fileName);
        if(mFile.getSize()!=0){ //File Null Check
            if(! file.exists()){
                if(file.getParentFile().mkdirs()){
                    file.createNewFile();
                }
            }
            mFile.transferTo(new File(ARTICLE_IMAGE_REPO + "\\" + "temp" + "\\" + imageFileName));
        }
    }
}
```

업로드한 파일 이름을 얻은 후 반환합니다.

30.5 새 글 추가하기

3. Service 클래스에서는 컨트롤러에서 전달된 articleMap을 다시 DAO의 insertNewArticle() 메서드 인자로 전달합니다.

코드 30-18 pro30/src/main/java/com/myspring/pro30/board/service/BoardServiceImpl.java

```
package com.myspring.pro30.board.service;

...

@Service("boardService")
@Transactional(propagation = Propagation.REQUIRED)
public class BoardServiceImpl implements BoardService{
    @Autowired
    BoardDAO boardDAO;

    ...

    @Override
    public int addNewArticle(Map articleMap) throws Exception{
        return boardDAO.insertNewArticle(articleMap);
    }
}
```

30.5 새 글 추가하기

4. DAO에서는 새 글에 대한 글 번호를 조회한 후 전달된 articleMap에 글 번호를 설정합니다. 그리고 insert() 메서드를 호출하면서 articleMap을 해당 id의 insert문으로 전달합니다.

코드 30-19 pro30/src/main/java/com/myspring/pro30/board/dao/BoardDAOImpl.java

```
package com.myspring.pro30.board.dao;
...
@Repository("boardDAO")
public class BoardDAOImpl implements BoardDAO {
    @Autowired
    private SqlSession sqlSession;
    ...
    @Override
    public int insertNewArticle(Map articleMap) throws DataAccessException {
        int articleNO = selectNewArticleNO();
        articleMap.put("articleNO", articleNO);
        sqlSession.insert("mapper.board.insertNewArticle", articleMap);
        return articleNO;
    }

    private int selectNewArticleNO() throws DataAccessException {
        return sqlSession.selectOne("mapper.board.selectNewArticleNO");
    }
    ...
}
```

새 글에 대한 글 번호를 가져옵니다.

글 번호를 articleMap에 저장합니다.

id에 대한 insert문을 호출하면서 articleMap을 전달합니다.

새 글 번호를 가져옵니다.

30.5 새 글 추가하기

• 30.5.3 JSP 구현하기

1. 글목록창에서 회원 로그인 상태(`isLogOn`)를 함수 인자 값으로 미리 저장해 놓습니다. 글쓰기를 클릭하면 자바스크립트 함수를 호출하면서 로그인 상태 여부에 따라 각각 다른 요청을 수행합니다.

코드 30-20 pro30/src/main/webapp/WEB-INF/views/board/listArticles.jsp

```
...
<script>
function fn_articleForm(isLogOn,articleForm,loginForm){
    if(isLogOn != '' && isLogOn != 'false'){
        location.href=articleForm;
    }else{
        alert("로그인 후 글쓰기가 가능합니다.")
        location.href=loginForm+'?action=/board/articleForm.do';
    }
}
</script>
...
<a class="cls1"
    href="javascript:fn_articleForm( ${isLogOn}', '${contextPath}/board/articleForm.do',
    '${contextPath}/member/loginForm.do')"><p class="cls2">글쓰기</p></a>
</body>
</html>
```

로그인 상태이면 글쓰기창으로 이동합니다.

로그아웃 상태이면 action 값으로 다음에 수행할 URL인 /board/articleForm.do를 전달하면서 로그인창으로 이동합니다.

현재 로그인 상태를 함수 인자로 미리 전달합니다.

로그인 상태일 경우 이동할 글쓰기창 요청 URL을 인자로 전달합니다.

로그인 상태가 아닐 경우 로그인창 요청 URL을 전달합니다.

30.5 새 글 추가하기

2. 글쓰기창에서는 글 정보와 함께 이미지 파일을 첨부하여 컨트롤러로 전송합니다.

코드 30-21 pro30/src/main/webapp/WEB-INF/views/board/articleForm.jsp

```
...
</script>
<title>글쓰기창</title>
</head>
<body>
<h1 style="text-align:center">글쓰기</h1>
<form name="articleForm" method="post"
      action="${contextPath}/board/addNewArticle.do"
      enctype="multipart/form-data">
<table border="0" align="center">
  <tr>
    <td align="right"> 작성자</td>
    <td colspan=2 align="left">
      <input type="text" size="20" maxlength="100"
        value="${member.name}" readonly/>
    </td>
  </tr>
</table>
...
```

로그인하면 작성자의 이름을 표시합니다.

30.5 새 글 추가하기

3. 브라우저로 요청하여 글목록창을 나타낸 후 글쓰기를 클릭합니다.

글번호	작성자	제목	작성일
1	hong	상품평	2018. 9. 18.
2	hong	상품평가	2018. 9. 18.
3	hong	상품 주문이 늦어요.	2018. 9. 18.
4	lee	[답변] 상품평입니다.	2018. 9. 19.
5	lee	[답변] 222	2018. 9. 19.
6	lee	[답변] 죄송합니다.	2018. 9. 18.
7	lee	[답변] 테스트글	2018. 9. 19.
8	lee	[답변] 사용 후기입니다.	2018. 9. 19.
9	hong	상품평입니다.	2018. 9. 18.
10	hong	최길동글입니다.	2018. 9. 18.
11	kim	김유신입니다.	2018. 9. 18.
12	lee	[답변] 이용 후기입니다.	2018. 9. 18.
13	hong	안녕하세요	2018. 9. 18.
14	lee	[답변] 상품후기입니다..	2018. 9. 18.
15	hong	[답변] 답변입니다.	2018. 9. 18.
16	lee	[답변] 답변입니다.	2018. 9. 18.
17	hong	테스트글입니다.	2018. 9. 18.

글쓰기

30.5 새 글 추가하기

4. 로그인 상태가 아닐 경우 로그인창으로 이동하여 로그인을 클릭합니다.

아이디	비밀번호
<input type="text" value="hong"/>	<input type="password" value="...."/>
<input type="button" value="로그인"/>	<input type="button" value="다시입력"/>

5. 로그인에 성공하면 다시 글쓰기창으로 이동하여 글 정보를 입력하고 글쓰기를 클릭합니다.

글쓰기

작성자

글제목:

글내용:

새 상품이 곧 출시됩니다.
기대하세요~

이미지파일 첨부: duke.png



30.5 새 글 추가하기

6. 다음과 같이 새 글 제목이 목록에 나타납니다

글번호	작성자	제목	작성일
1	hong	상품정보입니다.	2018. 10. 15.
2	hong	상품평	2018. 9. 18.
3	hong	상품평가	2018. 9. 18.
4	hong	상품 주문이 늦어요.	2018. 9. 18.
5	lee	[답변] 상품평입니다.	2018. 9. 19.
6	lee	[답변] 222	2018. 9. 19.
7	lee	[답변] 죄송합니다.	2018. 9. 18.
8	lee	[답변] 테스트글	2018. 9. 19.
9	lee	[답변] 사용 후기입니다.	2018. 9. 19.
10	hong	상품평입니다.	2018. 9. 18.
11	hong	최길동글입니다.	2018. 9. 18.
12	kim	김유신입니다.	2018. 9. 18.
13	lee	[답변] 이용 후기입니다.	2018. 9. 18.
14	hong	안녕하세요	2018. 9. 18.
15	lee	[답변] 상품후기입니다..	2018. 9. 18.
16	hong	[답변] 답변입니다.	2018. 9. 18.
17	lee	[답변] 답변입니다.	2018. 9. 18.
18	hong	테스트글입니다.	2018. 9. 18.

글쓰기

30.6 글상세 창 구현하기

1. 매퍼 파일 board.xml에 전달된 글 번호에 대해 글 정보를 조회하는 SQL문을 추가합니다.

코드 30-22 pro30/src/main/resources/mybatis/mappers/board.xml

```
...
<select id="selectArticle" resultType="articleVO" parameterType="int">
  <![CDATA[
    SELECT * from t_board
    where articleNO = #{articleNO}
  ]]>
</select>
...
```

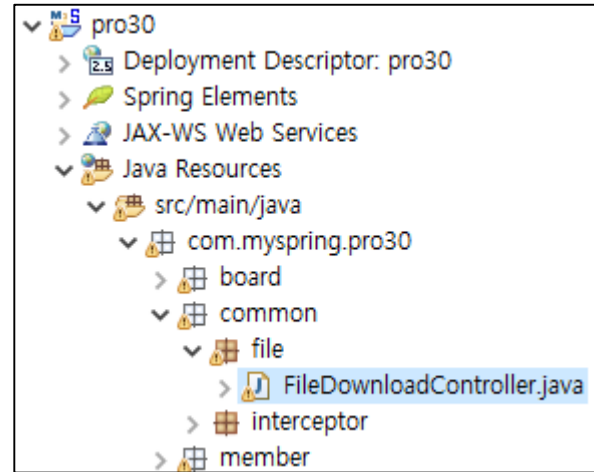
2. 글상세창(viewArticle.jsp)을 나타낼 타일즈 기능을 설정합니다

코드 30-23 pro30/src/main/resources/tiles/tiles_board.xml

```
...
<definition name="/board/viewArticle" extends="baseLayout">
  <put-attribute name="title" value="글상세창" />
  <put-attribute name="body" value="/WEB-INF/views/board/viewArticle.jsp" />
</definition>
...
```

30.6 글상세 창 구현하기

3. 첨부 파일을 표시할 파일 다운로드 컨트롤러인 FileDownloadController를 common/file 패키지에 구현합니다.



30.6 글상세 창 구현하기

4. 28장의 FileDownloadController 클래스를 복사해 붙여 넣은 후 다음과 같이 수정합니다.

코드 30-24 pro30/src/main/java/com/myspring/pro30/common/file/FileDownloadController.java

```
package com.myspring.pro30.common.file;

...
@Controller
public class FileDownloadController {
    private static final String ARTICLE_IMAGE_REPO = "C:\\board\\article_image";
    @RequestMapping("/download.do") 이미지 파일 이름을 바로 설정합니다.
    protected void download(@RequestParam("imageFileName") String imageFileName,
                           @RequestParam("articleNO") String articleNO,
                           HttpServletResponse response)throws Exception {
        OutputStream out = response.getOutputStream();
        String downFile = ARTICLE_IMAGE_REPO + "\\ "+articleNO+"\\ "+ imageFileName;
        File file = new File(downFile); 글 번호와 파일 이름으로 다운로드할
        ... 파일 경로를 설정합니다.
    }
}
```

30.6 글상세 창 구현하기

5. 글목록창에서 전달된 글 번호를 이용하여 해당 글 정보를 조회합니다

코드 30-25 pro30/src/main/java/com/myspring/pro30/board/controller/BoardControllerImpl.java

```
package com.myspring.pro30.board.controller;

...

@Controller("boardController")
public class BoardControllerImpl implements BoardController{
    private static final String ARTICLE_IMAGE_REPO = "C:\\board\\article_image";

    @Autowired
    BoardService boardService;

    @Autowired
    ArticleVO articleVO;

    ...

    @RequestMapping(value="/board/viewArticle.do", method = RequestMethod.GET)
    public ModelAndView viewArticle(@RequestParam("articleNO") int articleNO,
                                    HttpServletRequest request,
                                    HttpServletResponse response) throws Exception{
        String viewName = (String)request.getAttribute("viewName");
        articleVO=boardService.viewArticle(articleNO);
        ModelAndView mav = new ModelAndView();
        mav.setViewName(viewName);
        mav.addObject("article", articleVO);
        return mav;
    }

    ...
}
```

조회할 글 번호를 가져옵니다.

조회한 글 정보를 articleVO에 설정합니다.

30.6 글상세 창 구현하기

6. Service 클래스와 DAO 클래스의 메서드는 제공하는 소스 파일을 참고하기 바랍니다.
7. 마지막으로 글 상세 정보를 표시할 JSP 파일을 다음과 같이 작성합니다. 단, 자신이 작성한 글만 수정, 삭제를 표시합니다.

코드 30-26 pro30/src/main/webapp/WEB-INF/views/board/viewArticle.jsp

```
...
<tr id="tr_btn">
  <td colspan="2" align="center">
    <c:if test="${member.id == article.id}">
      <input type="button" value="수정하기" onClick="fn_enable(this.form)">
      <input type="button" value="삭제하기" onClick="fn_remove_article(
        '${contextPath}/board/removeArticle.do', ${article.articleNO})">
    </c:if>
    <input type="button" value="리스트로 돌아가기" onClick="backToList(this.form)">
    <input type="button" value="답글쓰기" onClick="fn_reply_form(
      ${contextPath}/board/replyForm.do', ${article.articleNO})">
  </td>
</tr>
</table>
...
```

로그인 ID가 작성자 ID와 같은 경우에만 수정하기, 삭제하기 버튼이 표시됩니다.

30.6 글상세 창 구현하기

8. 다음은 실행 결과입니다. 로그인하지 않았을 때와 로그인했을 때의 결과가 어떻게 다른지 비교해 보세요.

글번호	18
작성자 아이디	hong
제목	상품정보입니다.
내용	새 상품이 곧 출시됩니다. 기대하세요~
이미지	
등록일자	2018. 10. 15.

[파일 선택](#) [선택된 파일 없음](#)

[리스트로 돌아가기](#) [답글쓰기](#)

30.6 글상세 창 구현하기

로그인했을 때: 수정하기와 삭제하기 버튼이 보임

글번호	18
작성자 아이디	hong
제목	상품정보입니다.
내용	새 상품이 곧 출시됩니다. 기대하세요~
이미지	
등록일자	2018. 10. 15.

[수정하기](#) [삭제하기](#) [리스트로 돌아가기](#) [답글쓰기](#)

30.7 글 수정하기

1. 매퍼 파일에 update문을 추가합니다. if문을 사용하여 글 수정 시 이미지를 수정한 경우만 이미지 파일 이름을 업데이트하도록 지정합니다.

코드 30-27 pro30/src/main/resources/mybatis/mappers/board.xml

```
...
<update id="updateArticle" parameterType="java.util.Map">
  update t_board
  set title=#{title}, content=#{content}
  <if test="imageFileName!='' and imageFileName!=null">
    ,imageFileName=#{imageFileName}
  </if>
  where articleNO=#{articleNO}
</update>
...
```

Map으로 글 정보를 가져옵니다.

이미지를 수정한 경우에만 이미지 파일 이름을 수정합니다.

30.7 글 수정하기

2. 자바 파일을 작성할 차례입니다.

코드 30-28 pro30/src/main/java/com/myspring/pro30/board/controller/BoardControllerImpl.java

```
...
@RequestMapping(value="/board/modArticle.do", method = RequestMethod.POST)
@ResponseBody
public ResponseEntity modArticle(MultipartHttpServletRequest multipartRequest,
                                HttpServletResponse response) throws Exception{
    multipartRequest.setCharacterEncoding("utf-8");
    Map<String, Object> articleMap = new HashMap<String, Object>();
    Enumeration enu=multipartRequest.getParameterNames();
    while(enu.hasMoreElements()){
        String name=(String)enu.nextElement();
        String value=multipartRequest.getParameter(name);
        articleMap.put(name,value);
    }

    String imageFileName= upload(multipartRequest);
    HttpSession session = multipartRequest.getSession();
    MemberVO memberVO = (MemberVO) session.getAttribute("member");
    String id = memberVO.getId();
    articleMap.put("id", id);
    articleMap.put("imageFileName", imageFileName);
    String articleNO=(String)articleMap.get("articleNO");
    String message;
    ResponseEntity resEnt=null;
    HttpHeaders responseHeaders = new HttpHeaders();
    responseHeaders.add("Content-Type", "text/html; charset=utf-8");
    try {
        boardService.modArticle(articleMap);
```

30.7 글 수정하기

```
if(imageFileName!=null && imageFileName.length()!=0) {  
    File srcFile = new File(ARTICLE_IMAGE_REPO+"\\temp"+"\\ "+imageFileName);  
    File destDir = new File(ARTICLE_IMAGE_REPO+"\\ "+articleNO);  
    FileUtils.moveFileToDirectory(srcFile, destDir, true);  
      
      
    String originalFileName = (String)articleMap.get("originalFileName");  
    File oldFile = new File(ARTICLE_IMAGE_REPO+"\\ "+articleNO+"\\ "+originalFileName);  
    oldFile.delete();  
}  
...  
}  
return resEnt;  
}  
...
```

새로 첨부한 파일을 폴더로 이동합니다.

기존 파일을 삭제합니다.

3. Service 클래스와 DAO 클래스는 이 책에서 제공하는 소스 코드를 참고하기 바랍니다.

30.7 글 수정하기

4. 17장에서 글상세창을 구현할 때 사용한 viewArticle.jsp를 복사해서 붙여 넣습니다. 그리고 첨부 파일이 없는 글을 수정할 때는 파일 업로드 기능이 표시되도록 수정해야 합니다

코드 30-29 pro30/src/main/webapp/WEB-INF/views/board/viewArticle.jsp

```
...
<c:choose>
  <c:when test="${not empty article.imageFileName && article.imageFileName!='null' }">
    <tr>
      <td width="150" align="center" bgcolor="#FF9933" rowspan="2">
        이미지
      </td>
      <td>
        <input type="hidden" name="originalFileName"
              value="${article.imageFileName}" />
        <br>
      </td>
    </tr>
    <tr>
      <td>
        <input type="file" name="imageFileName" id="i_imageFileName" disabled
              onchange="readURL(this);" />
      </td>
    </tr>
  </c:when>
  <c:otherwise>
    <tr id="tr_file_upload" >
      <td width="150" align="center" bgcolor="#FF9933" rowspan="2">
```

30.7 글 수정하기

```
        이미지
    </td>
    <td>
        <input type= "hidden" name="originalFileName" value="${article.
                                                    imageFileName }" />

    </td>
</tr>
<tr>
    <td ></td>
    <td>
        <img id="preview" /><br>
        <input type="file" name="imageFileName " id="i_imageFileName" disabled
            onchange="readURL(this);" />

    </td>
</tr>
</c:otherwise>
</c:choose>
...
```

첨부 파일이 없는 글을 수정할 때는
파일 업로드가 표시되게 합니다.

30.7 글 수정하기

5. 다음은 실행 결과입니다. 글상세창에서 수정하기를 클릭하면 글을 수정할 수 있는 상태로 바뀝니다.
글 내용과 이미지를 수정한 후 수정반영하기를 클릭합니다

글번호	18
작성자 아이디	hong
제목	사용 후기입니다.
내용	상품 질이 좋아요. 수고하세요.
이미지	
등록일자	2018. 10. 15.
<div>파일 선택 선택된 파일 없음</div>	
<div>수정반영하기 취소</div>	

30.7 글 수정하기

6. 그러면 다음과 같이 수정된 글 정보를 테이블에 반영하고 새로운 이미지를 업로드한 후 다시 글상세창을 나타냅니다.

글번호	18
작성자 아이디	hong
제목	사용 후기입니다.
내용	상품 질이 좋아요. 수고하세요. 하이팅입니다!!!
이미지	
등록일자	2018. 10. 15.

30.8 글 삭제하기

1. 매퍼 파일에 글 번호를 가져와 관련된 자식 글까지 삭제하는 delete문을 추가합니다.

코드 30-30 pro30/src/main/resources/mybatis/mappers/board.xml

```
...
<delete id="deleteArticle" parameterType="int">
  <![CDATA[
    delete from t_board
    where articleNO in (
      SELECT articleNO FROM t_board
      START WITH articleNO = #{articleNO}
      CONNECT BY PRIOR articleNO = parentNO )
  ]]>
</delete>
...
```

글 번호를 가져옵니다.

30.8 글 삭제하기

2. 컨트롤러에서는 글 번호를 가져와 해당 글을 삭제합니다. 그리고 해당되는 이미지가 저장된 폴더도 함께 삭제합니다

코드 30-31 pro30/src/main/java/com/myspring/pro30/board/controller/BoardControllerImpl.java

```
...
@Override
@RequestMapping(value="/board/removeArticle.do" ,method = RequestMethod.POST)
@ResponseBody
public ResponseEntity removeArticle(@RequestParam("articleNO") int articleNO
                                     HttpServletRequest request,
                                     HttpServletResponse response) throws Exception{
    response.setContentType("text/html; charset=UTF-8");
    String message;
    ResponseEntity resEnt=null;
    HttpHeaders responseHeaders = new HttpHeaders();
    responseHeaders.add("Content-Type", "text/html; charset=utf-8");
    try {
        boardService.removeArticle(articleNO);
        File destDir = new File(ARTICLE_IMAGE_REPO+"\\\\"+articleNO);
        FileUtils.deleteDirectory(destDir);
        ...
    }catch(Exception e) {
        ...
    }
    return resEnt;
}
...
```

삭제할 글 번호를 가져옵니다.

글 번호를 전달해서 글을 삭제합니다.

글에 첨부된 이미지 파일이 저장된 폴더도 삭제합니다.

30.8 글 삭제하기

3. Service 클래스와 DAO 클래스는 이 책에서 제공하는 소스 파일을 참고하기 바랍니다.

4. 다음은 실행 결과입니다. 삭제하기를 클릭합니다.

글번호	18
작성자 아이디	hong
제목	사용 후기입니다.
내용	상품 질이 좋아요. 수고하세요. 하이팅입니다!!!
이미지	
등록일자	2018. 10. 15.
<input type="button" value="수정하기"/> <input type="button" value="삭제하기"/> <input type="button" value="리스트로 돌아가기"/> <input type="button" value="답글쓰기"/>	

30.8 글 삭제하기

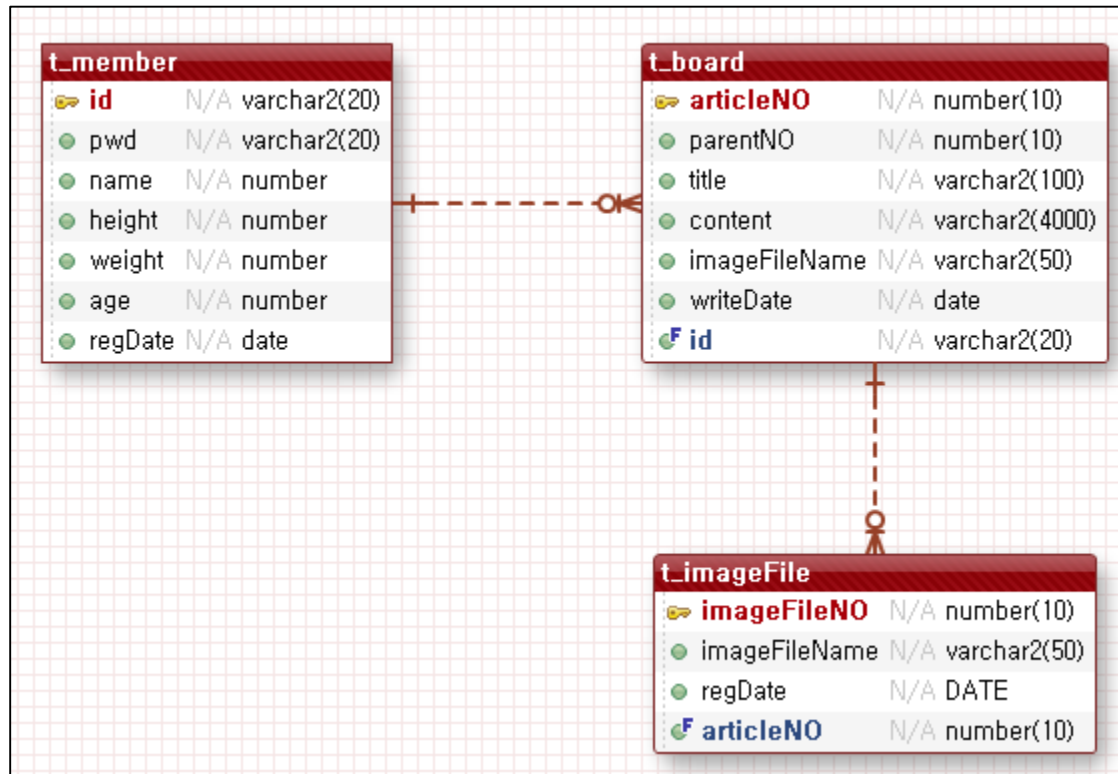
5. 해당 글을 삭제한 후 글목록창을 다시 표시합니다.

글번호	작성자	제목	작성일
1	hong	상품평	2018. 9. 18.
2	hong	상품평가	2018. 9. 18.
3	hong	상품 주문이 늦어요.	2018. 9. 18.
4	lee	[답변] 상품평입니다.	2018. 9. 19.
5	lee	[답변] 222	2018. 9. 19.
6	lee	[답변] 죄송합니다.	2018. 9. 18.
7	lee	[답변] 테스트글	2018. 9. 19.
8	lee	[답변] 사용 후기입니다.	2018. 9. 19.
9	hong	상품평입니다.	2018. 9. 18.
10	hong	최길동글입니다.	2018. 9. 18.
11	kim	김유신입니다.	2018. 9. 18.
12	lee	[답변] 이용 후기입니다.	2018. 9. 18.
13	hong	안녕하세요	2018. 9. 18.
14	lee	[답변] 상품후기입니다..	2018. 9. 18.
15	hong	[답변] 답변입니다.	2018. 9. 18.
16	lee	[답변] 답변입니다.	2018. 9. 18.
17	hong	테스트글입니다.	2018. 9. 18.

글쓰기

30.9 새 글 추가 시 여러 이미지 파일 첨부

이미지 정보 테이블 생성



30.9 새 글 추가 시 여러 이미지 파일 첨부

이미지 정보 테이블 구조

no	컬럼이름	속성 이름	자료형	크기	유일키 여부	NULL 여부	키	기본값
1	imageFileNO	이미지 파일 번호	number	10	Y	N	기본키	
2	imageFileName	이미지 파일 이름	varchar2	50	N	N		
3	regData	등록일자	date		N	N		sysdate
4	articleNO	글 번호	number	10	Y	N	참조키	

1. SQL Developer를 이용해 이미지 테이블을 생성합니다. SQL문 실행 시 마지막에 게시판 테이블의 articleNO 컬럼을 참조하는 구문을 추가합니다

코드 30-32 이미지 저장 테이블 생성 SQL문

```
CREATE TABLE t_imageFile (
  imageFileNO number(10),
  imageFileName varchar2(50),
  regDate date default sysdate,
  articleNO number(10),
  CONSTRAINT FK_ARTICLENO FOREIGN KEY(articleNO)
  REFERENCES t_board(articleNO) ON DELETE CASCADE
);
```

게시판 글을 삭제할 경우 해당 글 번호를 참조하는 이미지 정보도 자동으로 삭제됩니다.

30.9 새 글 추가 시 여러 이미지 파일 첨부

2. 매퍼 파일에 다음과 같이 SQL문을 추가합니다. 새 글 추가 시 여러 개의 이미지 파일을 첨부할 수 있기 때문에 insert문에서 <foreach> 태그를 이용해 한꺼번에 추가할 수 있게 합니다

코드 30-33 pro30/src/main/resources/mybatis/mappers/board.xml

```
...
<insert id="insertNewArticle" parameterType="java.util.Map">
  <![CDATA[
    insert into t_board(articleNO, id, title, content, imageFileName)
    values(#{articleNO}, #{id}, #{title}, #{content}, null)
  ]]>
</insert>

<insert id="insertNewImage" parameterType="java.util.Map">
  <foreach item="item" collection="list" open="INSERT ALL"
    separator=" " close="SELECT * FROM DUAL" >
    INTO t_imageFile(imageFileNO, imageFileName, articleNO, regDate)
    VALUES (#{item.imageFileNO}, #{item.imageFileName}, #{item.articleNO}, sysdate)
  </foreach>
</insert>
<select id="selectNewImageFileNO" resultType="int" >
  <![CDATA[
    SELECT nvl(max(imageFileNO),0) from t_imageFile
  ]]>
</select>
...
```

imageFileName 컬럼은 더 이상 사용하지 않으므로 null을 입력합니다.

한꺼번에 여러 개의 레코드를 추가합니다.

새로 추가할 이미지 정보의 이미지 번호를 가져옵니다.

30.9 새 글 추가 시 여러 이미지 파일 첨부

3. 이번에는 자바 파일을 수정할 차례입니다.

코드 30-34 pro30/src/main/java/com/myspring/pro30/board/controller/BoardControllerImpl.java

```
...
@Override
@RequestMapping(value="/board/addNewArticle.do", method = RequestMethod.POST)
@ResponseBody
public ResponseEntity addNewArticle(MultipartHttpServletRequest multipartRequest,
                                     HttpServletResponse response) throws Exception {
    ...
    List<String> fileList = upload(multipartRequest);
    List<ImageVO> imageFileList = new ArrayList<ImageVO>();
    if(fileList!= null && fileList.size()!=0) {
        for(String fileName : fileList) {
            ImageVO imageVO = new ImageVO();
            imageVO.setImageFileName(fileName);
            imageFileList.add(imageVO);
        }
        articleMap.put("imageFileList", imageFileList);
    }
    ...
}
```

첨부한 이름을 fileList로 반환합니다.

전송된 이미지 정보를 ImageVO 객체의 속성에 차례대로 저장한 후 imageFileList에 다시 저장합니다.

imageFileList를 다시 articleMap에 저장합니다.

30.9 새 글 추가 시 여러 이미지 파일 첨부

```
try {  
    int articleNO = boardService.addNewArticle(articleMap);  
    if(imageFileList!=null && imageFileList.size()!=0) {  
        for(ImageVO imageVO:imageFileList) {  
            imageFileName = imageVO.getImageFileName();  
            File srcFile = new File(ARTICLE_IMAGE_REPO+"\\temp"+"\\ "+imageFileName);  
            File destDir = new File(ARTICLE_IMAGE_REPO+"\\ "+articleNO);  
            //destDir.mkdirs();  
            FileUtils.moveFileToDirectory(srcFile, destDir,true);  
        }  
    }  
    ...  
}catch(Exception e) {  
    if(imageFileList!=null && imageFileList.size()!=0) {  
        for(ImageVO imageVO:imageFileList) {  
            imageFileName = imageVO.getImageFileName();  
            File srcFile = new File(ARTICLE_IMAGE_REPO+"\\temp"+"\\ "+imageFileName);  
            srcFile.delete();  
        }  
    }  
    ...  
}  
return resEnt;  
}
```

articleMap을 서비스 클래스로 전달합니다.

첨부한 이미지들을 for문을 이용해 업로드합니다.

오류 발생 시 temp 폴더의 이미지들을 모두 삭제합니다.

30.9 새 글 추가 시 여러 이미지 파일 첨부

```
...  
private List<String> upload(MultipartHttpServletRequest multipartRequest)  
throws Exception{  
    List<String> fileList= new ArrayList<String>();  
    Iterator<String> fileNames = multipartRequest.getFileNames();  
    while(fileNames.hasNext()){  
        String fileName = fileNames.next();  
        MultipartFile mFile = multipartRequest.getFile(fileName);  
        String originalFileName=mFile.getOriginalFilename();  
        fileList.add(originalFileName);  
        File file = new File(ARTICLE_IMAGE_REPO + "\\\" + fileName);  
        if(mFile.getSize()!=0){  
            if(! file.exists()){  
                if(file.getParentFile().mkdirs()){  
                    file.createNewFile();  
                }  
            }  
            mFile.transferTo(new File(ARTICLE_IMAGE_REPO + "\\\" + "temp"  
                                     + "\\\" + originalFileName));  
        }  
    }  
    return fileList;  
}
```

이미지 파일 이름이 저장된 List를 반환합니다.

첨부한 이미지 파일의 이름들을 차례대로 저장합니다.

30.9 새 글 추가 시 여러 이미지 파일 첨부

4. Service 클래스와 DAO 클래스를 각각 다음과 같이 작성합니다.

코드 30-35 pro30/src/main/java/com/myspring/pro30/board/service/BoardServiceImpl.java

```
...
@Override
public int addNewArticle(Map articleMap) throws Exception{
    int articleNO = boardDAO.insertNewArticle(articleMap);
    articleMap.put("articleNO", articleNO);
    boardDAO.insertNewImage(articleMap);
    return articleNO;
}
...
```

글 정보를 저장한 후 글 번호를 가져옵니다.

글 번호를 articleMap에 저장한 후 이미지 정보를 저장합니다.

30.9 새 글 추가 시 여러 이미지 파일 첨부

코드 30-36 pro30/src/main/java/com/myspring/pro30/board/service/BoardDAOImpl.java

...

```
public int insertNewArticle(Map articleMap) throws DataAccessException {  
    int articleNO = selectNewArticleNO();  
    articleMap.put("articleNO", articleNO);  
    sqlSession.insert("mapper.board.insertNewArticle", articleMap);  
    return articleNO;  
}
```

글 정보를 게시판 테이블에 추가한 후 글 번호를 반환합니다.

@Override

```
public void insertNewImage(Map articleMap) throws DataAccessException {  
    List<ImageVO> imageFileList = (ArrayList)articleMap.get("imageFileList");  
    int articleNO = (Integer)articleMap.get("articleNO");  
    int imageFileNO = selectNewImageFileNO();  
    for(ImageVO imageVO : imageFileList){  
        imageVO.setImageFileNO(++imageFileNO);  
        imageVO.setArticleNO(articleNO);  
    }  
    sqlSession.insert("mapper.board.insertNewImage", imageFileList);  
}  
...  
private int selectNewImageFileNO() throws DataAccessException {  
    return sqlSession.selectOne("mapper.board.selectNewImageFileNO");  
}  
...
```

articleMap이 글 번호를

가져옵니다.

이미지 번호를

가져옵니다.
ImageVO 객체를 차례대로 가져와 이미지
번호와 글 번호 속성을 설정합니다.

30.9 새 글 추가 시 여러 이미지 파일 첨부

5. 이미지 테이블의 각 컬럼 이름에 대응하도록 ImageVO 클래스를 생성합니다.

코드 30-37 pro30/src/main/java/com/myspring/pro30/board/service/ImageVO.java

```
package com.myspring.pro30.board.vo;

...

public class ImageVO {
    private int imageFileNO;
    private String imageFileName;
    private Date regDate;
    private int articleNO;

    //각 속성에 대한 getter/setter
    ...
}
```

30.9 새 글 추가 시 여러 이미지 파일 첨부

6. 28.1절의 다중 파일 업로드 기능을 참고하여 새 글 등록창에 파일 업로드를 할 수 있는 기능을 다음과 같이 자바스크립트로 구현합니다

코드 30-38 pro30/src/main/webapp/WEB-INF/views/board/articleForm.jsp

```

...
var cnt=1;
function fn_addFile(){
    $("#d_file").append("<br>"+<input type='file' name='file'+cnt+' ' />");
    cnt++;
}
</script>
...
<tr>
    ...
    <td align="right">이미지파일 첨부</td>
    <td align="left"><input type="button" value="파일 추가" onClick="fn_addFile()" /></td>
    <td><div id="d_file"></div></td>
</tr>
...

```

파일 업로드 기능을 동적으로 추가합니다.

파일 추가를 클릭하면 <div>에 동적으로 파일 업로드가 추가됩니다.

30.9 새 글 추가 시 여러 이미지 파일 첨부

7. 브라우저에 요청하여 로그인한 후 다음과 같이 새 글을 작성합니다. 그리고 여러 개의 이미지 파일을 첨부한 후 글쓰기를 클릭하면 새 글이 등록됩니다.

글쓰기

작성자

글제목:

글내용:

사용 중인 상품 이미지를 첨부합니다.

이미지파일 첨부

파일 추가

파일 선택

duke.png

파일 선택

duke2.jpg

파일 선택

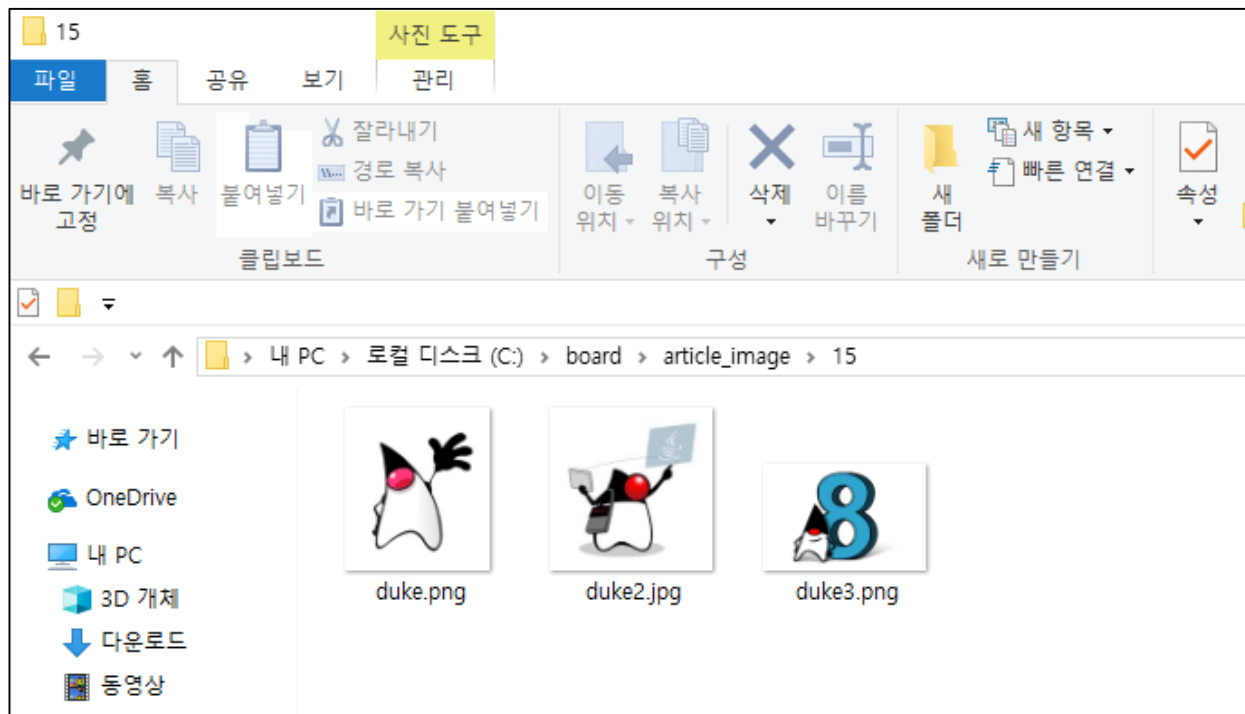
duke3.png

글쓰기

목록보기

30.9 새 글 추가 시 여러 이미지 파일 첨부

8. 글 번호에 해당하는 폴더를 열어 보면 첨부한 여러 개의 이미지 파일들이 실제로 업로드된것을 확인할 수 있습니다



30.10 글상세창에 여러 이미지 표시하기

1. modelConfig.xml에 ImageVO에 대한 alias를 설정합니다.

코드 30-39 pro30/src/main/resources/mybatis/model/modelConfig.xml

```
...  
<configuration>  
  <typeAliases>  
    <typeAlias type="com.myspring.pro30.member.vo.MemberVO" alias="memberVO" />  
    <typeAlias type="com.myspring.pro30.board.vo.ArticleVO" alias="articleVO" />  
    <typeAlias type="com.myspring.pro30.board.vo.ImageVO" alias="imageVO" />  
  </typeAliases>  
</configuration>
```

30.10 글상세창에 여러 이미지 표시하기

2. 이미지 파일 정보를 저장할 resultMap을 imageVO를 이용해 설정합니다.

코드 30-40 pro30/src/main/resources/mybatis/mappers/board.xml

...

```
<resultMap id="imgResult" type="imageVO">
  <result property="imageFileNO" column="imageFileNO" />
  <result property="articleNO" column="articleNO" />
  <result property="imageFileName" column="imageFileName" />
</resultMap>
```

...

이미지 파일 테이블에서 조회한 레코드를 저장하는 resultMap을 설정합니다.

```
<select id="selectImageFileList" resultMap="imgResult" parameterType="int">
```

```
<![CDATA[
```

```
  SELECT * from t_imageFile
  where articleNO = #{articleNO}
```

이미지 테이블에서 글 번호에 대한 이미지 파일 정보를 조회합니다.

```
]]>
```

```
</select>
```

...

30.10 글상세창에 여러 이미지 표시하기

3. JSP로 글 정보와 이미지 파일 정보를 한꺼번에 넘기도록 자바 파일을 수정하겠습니다.
articleMap에 각 정보를 key/value로 저장하여 JSP로 전달합니다.

코드 30-41 pro30/src/main/java/com/myspring/pro30/board/controller/BoardControllerImpl.java

```
...
@RequestMapping(value="/board/viewArticle.do", method = RequestMethod.GET)
public ModelAndView viewArticle(@RequestParam("articleNO") int articleNO,
                                HttpServletRequest request,
                                HttpServletResponse response) throws Exception{
    String viewName = (String)request.getAttribute("viewName");
    Map articleMap = boardService.viewArticle(articleNO);
    ModelAndView mav = new ModelAndView();
    mav.setViewName(viewName);
    mav.addObject("articleMap", articleMap);
    return mav;
}
...
```

서비스에서 조회한 글 정보와 이미지 파일 정보를 담은 Map을 가져옵니다.

articleMap을 JSP에 전달합니다.

30.10 글상세창에 여러 이미지 표시하기

4. 새 글과 이미지 파일 정보를 조회하도록 서비스 파일을 수정합니다.

코드 30-42 pro30/src/main/java/com/myspring/pro30/board/service/BoardServiceImpl.java

```
...
@Override
public Map viewArticle(int articleNO) throws Exception {
    Map articleMap = new HashMap();      글 정보를 조회합니다.
    ArticleVO articleVO = boardDAO.selectArticle(articleNO);
    List<ImageVO> imageFileList = boardDAO.selectImageFileList(articleNO);
    articleMap.put("article", articleVO);      이미지 파일 정보를 조회합니다.
    articleMap.put("imageFileList", imageFileList);
    return articleMap;      글 정보와 이미지 파일 정보를 Map에 담습니다.
}
...
```

5. BoardDAOImpl 클래스는 소스 파일을 참고하기 바랍니다.

30.10 글상세창에 여러 이미지 표시하기

6. JSP 파일에서는 이미지 파일 정보가 imageFileList로 전달되도록 설정합니다. <c:forEach>태그를 이용해 imageFileList에 저장된 ImageVO의 개수만큼 이미지를 표시합니다

코드 30-43 pro30/src/main/webapp/WEB-INF/views/viewArticle.jsp

```

...
<c:set var="article" value="${articleMap.article}" />
<c:set var="imageFileList" value="${articleMap.imageFileList}"/>
...
Map으로 넘어온 속성들의 이름을 다시 간단하게 설정합니다.
<c:if test="${not empty imageFileList && imageFileList != 'null'}">
    <c:forEach var="item" items="${imageFileList}" varStatus="status" >
        <tr>
            <td width="150" align="center" bgcolor="#FF9933" rowspan="2">
                이미지${status.count }
            </td>
            <td>
                <input type="hidden" name="originalFileName" value="${item.imageFileName }" />
                <br>
            </td>
        </tr>
        <tr>
            <td>
                <input type="file" name="imageFileName " id="i_imageFileName"
                    disabled onchange="readURL(this);" />
            </td>
        </tr>
    </c:forEach>
</c:if>
...
    
```

imageFileList가 비어 있지 않으면 이미지를 표시합니다.

<forEach> 태그를 이용해 이미지 개수만큼 반복해서 다운로드합니다.

30.10 글상세창에 여러 이미지 표시하기

7. 실행 결과를 보면 첨부한 세 개의 이미지가 나타나 있습니다

