

Java Web Programming 입문

(Servlet/JSP)

30일차

오늘의 키워드

▶ 게시판

◦ Java

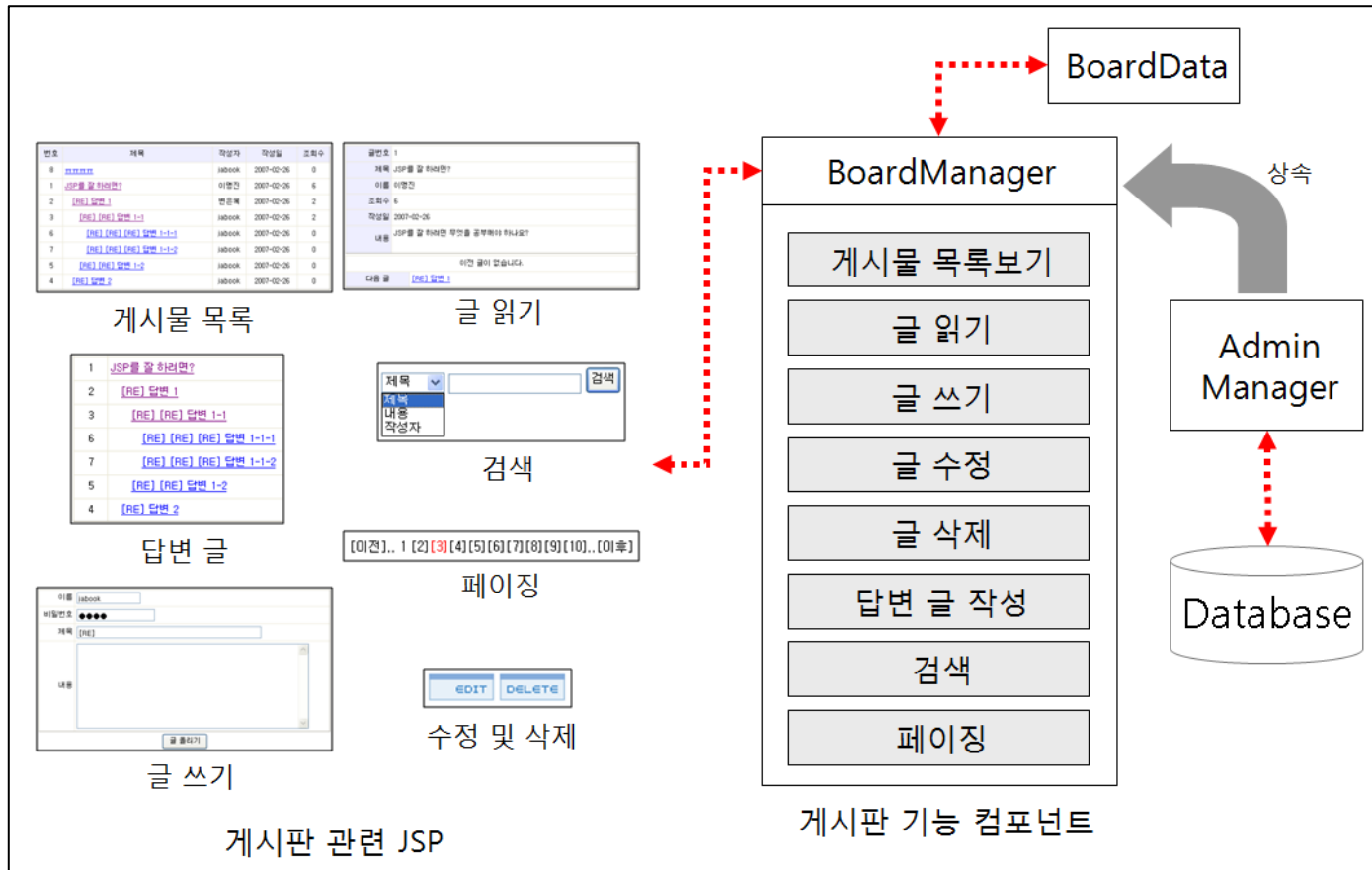
- 글 입력
- 글 읽기
- 글 수정
- 답변 글 작성
- 글 목록
- 페이징 (Paging)
- 글 검색
- 글 삭제

◦ JSP Page

- 글 목록
- 글 읽기
- 답변 글 쓰기
- 글 수정
- 글 삭제

게시판

▶ 게시판 Component



게시판

▶ 게시판 Component

◦ BoardManagerSkeletonCode.java

```
import java.util.*;
import java.sql.*;

public class BoardManager extends AdminManager{

    public static final int LINE_PER_PAGE    = 10; // 한페이지에 나타날 글의 수
    public static final int PAGES_PER_GROUP  = 10; // 한 화면에 나타낼 페이지수

    private int      recNum      = 0; // 전체 레코드 수
    private int      totalPage  = 0; // 총페이지 수
    private int      currentPage = 0; // 현재페이지
    private String    search     = ""; // 검색값을 위한 변수
    private String    text      = ""; // 검색결과를 위한 변수
    private boolean   success;    // 작업이 성공적으로 끝났는지를 확인하는 변수

    public int      getMaxNum      ( String boardName) { /* 데이터베이스에서 가장 큰 글 번호를 가져오는 메소드 */ }
    public void     writeNotice    ( String boardName, BoardData data ) { /* 새로운 글을 입력하기 위한 메소드 */ }
    public BoardData getNotice     ( String boardName, int num ) { /* 하나의 글을 반환하는 메소드 */ }
    public String   getPassword    ( String boardName, int num ) { /* 글의 비밀번호를 리턴하는 메소드 */ }
    public void     updateNotice   ( String boardName, BoardData data ) { /* 글을 수정하기 위한 메소드 */ }
    public void     replyNotice    ( String boardName, BoardData data ) { /* 답변글을 달기 위한 메소드 */ }
    public Vector   getPage        ( String boardName, int currentPage ) { /* 한 페이지의 글 목록을 Vector로 반환하는 메소드 */ }
    public String   getPaging      ( String boardName, int currentPage ) { /* 페이징을 위한 메소드 */ }
    public void     setSearch      ( String search ) { /* 어느 곳을 검색할 것인지 설정을 하는 부분 */ }
    public void     setText        ( String text ) { /* 검색값을 받아오는 부분 */ }
    public String   deleteNotice  ( String boardName, int num ) { /* 글을 삭제하기 위한 메소드 */ }
    public void     increaseCnt    ( String boardName, int num ) { /* 글의 조회수를 증가시키기 위한 메소드 */ }
    public int[]    getNextPrevNum ( String boardName, int num ) { /* 이전글과 다음 글의 번호를 구하기 위한 메소드 */ }
    public boolean  getSuccess     () { /* 작업수행의 결과를 리턴 */ }
    public void     setSuccess     ( boolean success ) { /* 작업수행의 결과를 설정 */ }
    public int      getRecNum      ( String boardName ) { /* 레코드 수를 반환하기 위한 메소드 */ }
}
```

게시판

▶ 게시판 Component

◦ BoardManager Class

- 글 읽기, 쓰기, 글 수정, 삭제 등의 기능 구현
- AdminManager Class 상속
 - AdminManager Class에 정의된 Database 관련 Method 사용

```
class BoardManager extends AdminManager { ... }
```

- 작업의 처리 성공 여부를 success라는 Flag Variable로 설정

```
private boolean success;  
public boolean getSuccess( ) { return success; }  
public void setSuccess(boolean success) { this.success = success; }
```

글 입력

- ▶ 새 글 올리기, 답 글 올리기
- ▶ 글 입력 시 입력된 내용은 Database에 저장
 - Database에 저장되기 전 점검 항목
 - 필수 입력 항목 검사
 - 가장 높은 글 번호 찾기

```
SELECT MAX(NUM) FROM BOARDNAME
```

- ▶ 가장 큰 글 번호 찾기
 - Parameter로 전달되 게시판에서 가장 큰 글 번호 반환

```
public int getMaxNum(String boardName) { ... }
```

- 가장 큰 글 번호를 찾아낼 SQL문 생성

```
String sql = "select max(Num) from " + boardName;
```

- adminExcuteQueryNum() Method를 실행하여 가장 큰 글 번호 반환

```
int maxNum = adminExecuteQueryNum(sql);  
return maxNum;
```

글 입력

▶ 게시물 등록

- 어떤 게시판에 올라갈 글인지?
- 올라갈 내용들은?

```
public void writeNotice(String boardName, BoardData data) { ... }
```

- 글 번호는?

```
noticeNum = this.getMaxNum(boardName) + 1;
```

- 글 저장을 위한 SQL문 생성

```
String sql = "Insert into " + boardName + " ";  
sql += "(num, name, ..., childCount)";  
sql += " values(" + noticeNum + ", '" + data.getName() + "', " + ... + ", 0)";
```

- Query 실행

```
adminExecuteUpdate(sql);
```

글 읽기

▶ 글 읽기

- getNotice() Method 사용
- 게시판 목록에서 원하는 글을 선택
 - 해당하는 내용을 Database에서 가져온 후 출력
 - 조건을 주고 Database Table에서 한 개 Row 만 가져오면 됨

◦ Method 선언

```
public BoardData getNotice(String boardName, int num) { ... }
```

◦ SQL문 생성

```
String sql = "select * from " + boardName + " where num=" + num;
```

◦ SQL 실행

```
Vector v = adminExecuteQuery(sql);
```

◦ 출력될 결과를 반환

```
BoardData data = (BoardData)v.elementAt(0);  
return data;
```


글 수정

▶ 글 수정

- 수정을 시도한 사용자가 해당 글 수정 권한이 있는지 검사
- 수정할 권한이 있다면 수정할 내용을 SQL 문 생성
- SQL 실행하여 Database에 저장된 글 수정

- 비밀번호 확인
 - Method 선언

```
public String getPassword(String boardName, int num) { ... }
```

- SQL문 생성

```
String sql = "select password from " + boardName + " where num=" + num;
```

- SQL문 실행 후 비밀번호 추출

```
String password = adminExecuteQueryString(sql);  
return password;
```

- 글 수정
 - Method 선언

```
public void updateNotice(String boardName, BoardData data) { ... }
```

- 수정 SQL 생성 및 실행

```
String sql = "update " + boardName + " set name='" + data.getName();  
sql += "', subject='" + data.getSubject() + "', content='" + data.getContent();  
sql += "' where num=" + data.getNum();  
adminExecuteUpdate(sql);  
this.setSuccess(true);
```

답변글 작성

- ▶ 답변 글과 관련된 Database Field
 - ref
 - 서로 관계 있는 글들을 묶어주는 번호
 - 원문 글 번호
 - step
 - 서로 관계 있는 글들의 순서를 나타내기 위해 사용
 - depth
 - 몇 단계의 답변 글인지를 표시
 - childCount
 - 해당 글에 달려있는 답변 글의 수
 - num
 - 글이 Database에 들어온 순서대로 부여 받은 글 번호

| num | subject | ref | step | depth | childCount | | |
|-----|-----------------------|-----|------|-------|------------|---|-----|
| 1 | JSP를 잘 하려면? | 1 | 0 | 0 | 6 | 0 | ref |
| 2 | [RE] 답변 1 | 1 | 1 | 1 | 4 | 1 | |
| 3 | [RE][RE] 답변 1-1 | 1 | 2 | 2 | 2 | 2 | |
| 6 | [RE][RE][RE] 답변 1-1-1 | 1 | 3 | 3 | 0 | 3 | |
| 7 | [RE][RE][RE] 답변 1-1-2 | 1 | 4 | 3 | 0 | 4 | |
| 5 | [RE][RE] 답변 1-2 | 1 | 5 | 2 | 0 | 5 | |
| 4 | [RE] 답변 2 | 1 | 6 | 1 | 0 | 6 | |

0 1 2 depth

답변글 작성

- ▶ 1번 글에 답변 글이 작성된다면

| num | subject | ref | step | depth | childCount |
|-----|-----------------|-----|------|-------|------------|
| 1 | JSP를 잘 하려면? | 1 | 0 | 0 | 2 |
| 2 | [RE] 답변 1 | 1 | 1 | 1 | 1 |
| 3 | [RE][RE] 답변 1-1 | 1 | 2 | 2 | 0 |

- 1번 글의 새로운 답변은 3번 글 아래에 작성
 - 1번 글의 ref, step, depth 를 전달
 - ref = 1, step = 0, depth = 0

```
SELECT min(step)
FROM 게시판 이름
WHERE ref = 원문의 ref
AND step > 원문의 step
AND depth <= 원문의 depth
```



```
SELECT min(step)
FROM freeboard
WHERE ref = 1
AND step > 0
AND depth <= 0
```

- 답변 글이 삽입되는 위치를 구하는 SQL
 - ref가 동일
 - step이 원문글보다 높고, 같은 단계이거나 상위 단계 답변이면서
 - 그 중 해당 Table에서 가장 작은 step을 가지는 곳
- 실행 결과가 0 이므로 답변글이 삽입될 위치는 step이 가장 큰 값의 다음 위치

답변글 작성

- 실행 결과가 0이므로 답변 글이 삽입될 위치는
 - step이 가장 큰 값의 다음 위치
 - 해당 ref의 최대값을 구해서 1을 더하여 답변 글이 들어갈 step 구함

```
( SELECT max(step) FROM freeboard WHERE ref = 1 ) + 1
```

- 다음 답변 글이 가질 num 값 추출

```
( SELECT max(Num) FROM freeboard ) + 1
```

- 최종적으로 답변 글이 가지는 값들
 - num = 4, ref = 1, step = 3, depth = 1

```
INSERT INTO FREEBOARD  
    ( num, ..., ref, step, depth, childCount )  
VALUES ( 4, ..., 1, 3, 1, 0 )
```

- 답변 글 수를 나타내는 childCount를 Update
 - 답변 글 수가 증가되어야 할 글을 찾아서 해당 글의 답변 글 수를 1 증가시킴

```
SELECT max(step)  
FROM freeboard  
WHERE ref = 원문의 ref  
AND step < 답변의 step  
AND depth = 답변의 _depth-1
```

```
UPDATE freeboard  
SET childCount = childCount + 1  
WHERE ref = 원문의 ref  
AND depth = 답변의 _depth-1  
AND step = 산출된 _max(step)
```

답변글 작성

▶ 3번 글에 답변 글이 작성된다면

| num | subject | ref | step | depth | childCount |
|-----|-----------------------|-----|------|-------|------------|
| 1 | JSP를 잘 하려면? | 1 | 0 | 0 | 5 |
| 2 | [RE] 답변 1 | 1 | 1 | 1 | 3 |
| 3 | [RE][RE] 답변 1-1 | 1 | 2 | 2 | 1 |
| 6 | [RE][RE][RE] 답변 1-1-1 | 1 | 3 | 3 | 0 |
| 5 | [RE][RE] 답변 1-2 | 1 | 4 | 2 | 0 |
| 4 | [RE] 답변 2 | 1 | 5 | 1 | 0 |

```
SELECT min(step)
FROM freeboard
WHERE ref = 1
AND step > 2
AND depth <= 2
```

- 5번, 6번 글 사이에 작성
- 3번 글에 대한 답변 글이므로 3번 글의 ref, step, depth 값을 넘김
 - ref = 1, step = 2, depth = 2
 -
- 결과 값이 0보다 큰 경우 그 값이 답변 글의 삽입 위치
 - 그 위치에 다른 답변 글들이 있을 경우 다른 글들을 아래로 밀어냄
 - 아래 답변 글들의 step을 1씩 증가시켜줌
- 답변 글이 가질 num 값 구한 후 글 삽입

```
UPDATE freeboard
SET step = step + 1
WHERE ref = 1
AND step >= 4
```

```
( SELECT max(num) FROM freeboard ) + 1
```

```
INSERT INTO freeboard(
    num, ..., ref, step, depth, childCount )
VALUES ( 7, ..., 1, 4, 3, 0 )
```

답변글 작성

▶ 답변 글 삽입 위치 계산

```
String sql      = "SELECT min(step) " +  
                  " FROM " + boardName +  
                  " WHERE ref      = " + data.getRef() +  
                  " AND depth <= " + data.getDepth() +  
                  " AND step > " + data.getStep();  
int mstep = adminExecuteQueryNum(sql);  
int instep = 0;
```

- step이 원문보다 높고,
◦ 같은 단계 혹은 상위 단계 답변이며,
◦ 그 중에서 가장 작은 step을 가지는 곳
◦ 0이거나 1보다 큰 값
 - 0 일 경우 해당 ref의 가장 마지막 step
 - 0보다 클 경우 답변 글의 사이에 위치

```
if(mstep > 0) {  
    // 답변 글들 사이에 위치  
} else {  
    // 동일한 ref의 가장 마지막 step에 삽입  
}
```

답변글 작성

- 0보다 클 경우 아래 답변 글들의 step을 1씩 증가

```
sql = "update " + boardName + " set step = step + 1 where ref = ";  
sql += data.getRef() + " and step >= " + mstep;  
adminExecuteUpdate(sql);  
instep = mstep;
```

- 0일 경우 답변 글이 삽입될 위치 계산 (step + 1)

```
sql = "select max(step) from " + boardName + " where ref = " + data.getRef();  
instep = adminExecuteQueryNum(sql) + 1;
```

▶ 글 번호 계산

```
int maxNum = getMaxNum(boardName) + 1;
```

```
public int getMaxNum(String boardName) {  
    String sql = "select max(Num) from " + boardName;  
    int maxNum = adminExecuteQueryNum(sql);  
    return maxNum;  
}
```

답변글 작성

▶ 글 삽입

```
int depth = data.getDepth();  
sql = ... + "num, ..., ref, step, depth, childCount" + ...;  
sql += ... + maxNum + "..." + data.getRef() + ", " +  
        instep + ", " + (++depth) + ", 0";  
adminExecuteUpdate(sql);
```

▶ 답변 글 수 증가

- 답변 글의 수를 나타내는 childCount를 Update
- 답변 글의 depth 만큼 반복

```
// 상위 글을 찾아서 childCount를 1씩 증가  
for(int r = depth - 1; r >= 0; r--) { ... }
```

▶ 답변 글이 달려있는 윗글 선택 후 childCount 1 증가

```
sql = "select max(step) from " + boardName + " where ref = " + data.getRef();  
sql += " and depth = " + r + " and step < " + instep;  
int max_Step = adminExecuteQueryNum(sql);
```

```
sql = "update " + boardName + " set childCount = childCount + 1 ";  
sql += "where ref = " + data.getRef() + " and depth = " + r + " and step = " + max_Step;  
adminExecuteUpdate(sql);
```


글 목록

▶ 글 목록

- Database에 저장된 글들을 최근 작성 순서로 정렬하여 화면에 보여줌
- Database에 저장된 글의 수가 너무 많을 경우
 - 저장된 모든 글을 한 화면에 보여줄 필요는 없음
 - 일정한 개수로 글 목록을 잘라내어 그 부분만 화면에 보여줌

▶ 글 목록 정렬

- 높은 ref로부터 낮은 ref 순으로 정렬
- ref가 동일할 경우 낮은 step에서 높은 step 순으로 정렬
- depth 값만큼 들여쓰기를 함으로써 어느 글의 답변 글인지 표현

| num | subject | ref | step | depth | childCount |
|-----|-------------|-----|------|-------|------------|
| 1 | 방가 방가~ | 1 | 0 | 0 | 0 |
| 2 | 축하합니다. | 2 | 0 | 0 | 1 |
| 3 | [RE] 축하합니다. | 2 | 1 | 1 | 0 |
| 4 | 안녕~ | 4 | 0 | 0 | 2 |
| 5 | [RE] 안녕 | 4 | 1 | 1 | 1 |
| 6 | [RE][RE] 축하 | 2 | 2 | 2 | 0 |
| 7 | 사랑합니다. | 7 | 0 | 0 | 0 |

데이터베이스에 저장된 순서



| num | subject | ref | step | depth | childCount |
|-----|-------------|-----|------|-------|------------|
| 7 | 사랑합니다. | 7 | 0 | 0 | 0 |
| 4 | 안녕~ | 4 | 0 | 0 | 2 |
| 5 | [RE] 안녕 | 4 | 1 | 1 | 1 |
| 2 | 축하합니다. | 2 | 0 | 0 | 1 |
| 3 | [RE] 축하합니다. | 2 | 1 | 1 | 0 |
| 6 | [RE][RE] 축하 | 2 | 2 | 2 | 0 |
| 1 | 방가 방가~ | 1 | 0 | 0 | 0 |

ref와 step 순으로 정렬된 순서

글 목록

▶ 출력 방법

- Database의 모든 Data를 불러와서 저장한 다음
 - 지정된 개수만큼 Page에 담아 보여주는 방법
- 글 목록의 번호를 미리 계산해서 산출된
 - 글 번호만을 데이터베이스로부터 가져와 출력해주는 방법

```
SELECT ... LIMIT 시작_레코드_번호, 레코드_수  
SELECT ... LIMIT 30, 10
```

```
public Vector getPage(String boardName, int currentPage) {  
    sql = "select * from " + boardName;  
    sql += " order by ref desc, step";  
    sql += " limit " + LINE_PER_PAGE * (currentPage - 1) + ", " + LINE_PER_PAGE;  
    return adminExecuteQuery(sql);  
}
```

페이징 (Paging)

▶ 페이징 (Paging)

- 하나의 페이지에 일정 개수만큼 글을 나누어 출력
 - 각 Page 사이의 이동이 가능하도록 해주는 기법
- 원하는 개수만큼의 글을 한 Page에서 보여줄 수 있음
 - 게시 순서에 따라 앞, 뒤 또는 원하는 Page 이동 가능
- 글 목록 하단에 Page간 이동을 해주는 Page Link 만들기
 - getPaging() Method

```
public String getPaging(String boardName, int currentPage) { ... }
```

- 출력될 시작 Page, 끝 Page 계산

```
this.currentPage = currentPage;
```

- 총 Page수

```
int recNum = this.getRecNum(boardName);  
this.totalPage = ((recNum - 1) / 10) + 1;
```

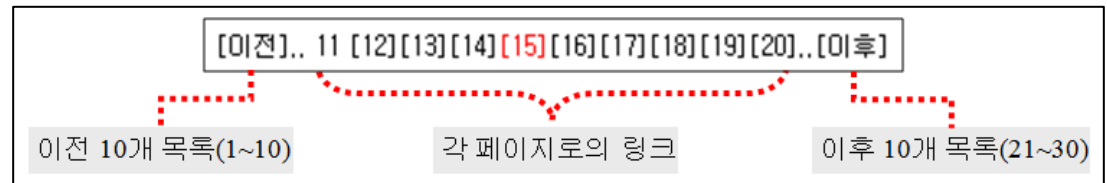
- 한 Page에 출력될 Page 링크 수 지정

```
PAGENUM = 10;
```

- 출력될 시작 Page 및 끝 Page 계산

```
startPage = ((currentPage - 1) / PAGENUM) * PAGENUM + 1;  
endpage = (((startpage - 1) + PAGENUM) / PAGENUM) * PAGENUM;
```

글 목록



▶ 페이징 (Paging)

◦ [이전] Button

- 현재 Page가 한 Page에 출력되는 Link의 수보다 크면
 - 이전 Page 목록으로 이동하는 Link 작성
- 크지 않다면
 - Link가 없는 “이전” 문자열만 출력

```
if(currentPage > PAGENUM) {  
    linkPage = startPage - 1;  
    strList = "<a href='" + list_url + "&page=" + linkPage + "'>[이전]</a>";  
} else {  
    strList = "[이전]";  
}
```

◦ Page Link

- 현재 Page가 위치한 Group 내의 Page를 Link 연결

```
linkPage = startPage;  
while(linkPage <= endPage) {  
    if(linkPage == currentPage){  
        strList += "&nbsp;" + currentPage + "&nbsp;";  
    } else {  
        strList += "<a href='" + boardLink + "&page=" + linkPage + "'>[" + linkPage + "]</a>";  
    }  
    linkPage++;  
}
```

글 목록

▶ 페이징 (Paging)

◦ [다음] Button

- 다음 Page 목록이 존재할 경우 Link 작성

```
if(totalPage > endPage) {  
    linkPage = endPage + 1;  
    strList += "<a href='" + boardLink + "&page=" + linkPage + "'>[이후]</a>";  
} else {  
    strList += "[이후]";  
}
```

글 검색

▶ 글 검색

- Database 저장된 글 중 특정어가 포함된 글을 찾아내는 기법
 - 찾아낸 글들을 정렬해서 화면에 보여줌
 - 글 제목, 글 내용, 글쓴이 등등
 - 글 목록 불러오기의 SQL문에 조건을 추가

```
public void setSearch(String search) {  
    this.search = search;  
}  
  
public void setText(String text) {  
    this.text = text;  
}
```

- search : 검색하려는 Field (Column)
- text : 검색어 설정

- 검색어가 있는 getPage() Method

```
public Vector getPage(String boardName, int currentPage) {  
    if(! (text.trim()).equals("")) {  
        // text - 검색 단어가 저장된 변수  
        // text 변수에 저장된 값을 포함하는 글의 목록 출력  
    } else {  
        // 검색 조건이 없는 일반적인 글 목록 출력  
    }  
    return adminExecuteQuery(sql);  
}
```

글 검색

- ▶ 검색된 글 목록 출력
 - 검색 조건이 있다면

```
sql = "select * from " + boardName;  
sql += " where upper(" + search + ") like upper('%" + text + "%')";  
sql += " order by ref desc, step";  
sql += " limit " + LINE_PER_PAGE * (currentPage - 1) + ", " + LINE_PER_PAGE;
```

- 없다면?

```
sql = "select * from " + boardName;  
sql += " order by ref desc, step";  
sql += " limit " + LINE_PER_PAGE * (currentPage - 1) + ", " + LINE_PER_PAGE;
```

글 삭제

▶ 글 삭제

- Database 영역에서 본다면 단순히 Delete
- 게시판 영역에서 본다면...
 - 답변이 있다면 글이 지우지 못하게 함
 - 지우되, '게시물이 삭제되었습니다' 표현
- 삭제 절차
 - 삭제할 글의 Data를 가져옴
 - 답변 글이 존재할 경우 글 삭제 불가
 - 답변 글이 없다면 글 삭제 가능
 - 답변 글의 원 글의 답변 글 수 1 감소

글 삭제

▶ 글 삭제

- deleteNotice() Method

```
public String deleteNotice(String boardName, int num) { ... }
```

- 삭제 글 정보 SELECT SQL

```
String sql = "select * from " + boardName + " where num=" + num;  
Vector v = adminExecuteQuery(sql);
```

```
BoardData data = (BoardData)v.elementAt(0);  
int ref = data.getRef();  
int step = data.getStep();  
int depth = data.getDepth();  
int childCount = data.getChildCount();
```

- 답변 글의 존재 여부 검사

```
if(childCount == 0) {  
    // 해당 글 삭제  
    // 삭제된 글을 포함하고 있는 상위 답글 수 감소  
} else {  
    // 답변 글이 존재하므로 삭제할 수 없음  
}
```

- 답변 글이 없다면 삭제

```
sql = "delete from " + boardName + " where num =" + num;  
adminExecuteUpdate(sql);
```

글 삭제

▶ 글 삭제

- deleteNotice() Method

```
for(int r = depth - 1; r >= 0; r--) {  
    sql = "select max(Step) from " + boardName + " where ref = " + ref ;  
    sql += " and depth = " + r + " and step < " + step;  
    int max_Step = adminExecuteQueryNum(sql);  
    sql = "update " + boardName + " set childCount = childCount-1 ";  
    sql += "where ref = " + ref + " and depth = " + r + " and step =" + max_Step;  
    adminExecuteUpdate(sql);  
}
```

▶ 글의 조회수 증가 Method

- 게시판 글 열람 시 글의 조회수 증가 increaseCnt()

```
String sql = "update " + boardName;  
sql += " set count = count + 1 where num=" + num;  
adminExecuteUpdate(sql);
```

▶ 이전 글, 다음 글 getNextPrevNum()

```
String sql = "select num from " + boardName;  
sql += " where num > " + num + " order by Num desc";  
nums[0] = adminExecuteQueryNum(sql);  
sql = "select num from " + boardName +  
    " where num < " + num;  
nums[1] = adminExecuteQueryNum(sql);  
return nums;
```

게시판

▶ BoardManager.java (시작)

```
import java.util.*;
import java.sql.*;

public class BoardManager extends AdminManager{
    public static final int LINE_PER_PAGE = 10; // 한페이지에 나타날 글의 수
    public static final int PAGES_PER_GROUP = 10; // 한 화면에 나타낼 페이지수
    private int recNum = 0; // 전체 레코드 수
    private int totalPage = 0; // 총페이지 수
    private int currentPage = 0; // 현재페이지
    private String search = ""; // 검색값을 위한 변수
    private String text = ""; // 검색결과를 위한 변수
    private boolean success; // 작업이 성공적으로 끝났는지를 확인하는 변수

    // 데이터베이스에서 가장 큰 글 번호를 가져오는 메소드
    public int getMaxNum(String boardName) {
        String sql = "select max(Num) from " + boardName;
        int maxNum = adminExecuteQueryNum(sql);
        return maxNum;
    }

    // 새로운 글을 입력하기 위한 메소드
    public void writeNotice(String boardName, BoardData data) {
        int noticeNum = this.getMaxNum(boardName) + 1;
        String sql = "Insert into " + boardName + " ";
        sql += "(num, name, subject, content, writeDate, password, ";
        sql += "count, ref, step, depth, childCount)";
        sql += " values(" + noticeNum + ", '" + data.getName() + "', '" +
            data.getSubject() + "', '" + data.getContent() + "', now(), '" +
            data.getPassword() + "', 0, " + noticeNum + ", 0, 0, 0)";
        adminExecuteUpdate(sql);
        this.setSuccess(true);
    }

    // 하나의 글을 반환하는 메소드
    public BoardData getNotice(String boardName, int num) {
        String sql = "select * from " + boardName + " where num=" + num;
        Vector v = adminExecuteQuery(sql);
        BoardData data = (BoardData)v.elementAt(0);
        return data;
    }
}
```

게시판

▶ BoardManager.java (계속)

```
// 글의 비밀번호를 리턴하는 메소드
public String getPassword(String boardName, int num) {
    String sql = "select password from " + boardName + " where num=" + num;
    String password = adminExecuteQueryString(sql);
    return password;
}

// 글을 수정하기 위한 메소드
public void updateNotice(String boardName, BoardData data) {
    String sql = "update " + boardName + " set name='" + data.getName();
    sql += "', subject='" + data.getSubject() + "', content='" + data.getContent();
    sql += "' where num=" + data.getNum();
    adminExecuteUpdate(sql);
    this.setSuccess(true);
}
```

```
// 한 페이지의 글목록을 Vector로 반환하는 메소드
public Vector getPage(String boardName, int currentPage) {
    String sql = null;
    if(! (text.trim()).equals("")){ //제목을 찾는 sql문
        sql = "select * from " + boardName;
        sql += " where upper(" + search + ") like upper('%" + text + "%')";
        sql += " order by ref desc, step limit " + LINE_PER_PAGE * (currentPage - 1) + ", " + LINE_PER_PAGE;
    }else {
        sql = "select * from " + boardName;
        sql += " order by ref desc, step limit " + LINE_PER_PAGE * (currentPage - 1) + ", " + LINE_PER_PAGE;
    }
    return adminExecuteQuery(sql);
}
```

게시판

▶ BoardManager.java (계속)

// 답변글을 달기 위한 메소드

```
public void replyNotice(String boardName, BoardData data) {
    String sql = "select min(step) from " + boardName + " where ref = " + data.getRef();
    sql += " and depth <= " + data.getDepth() + " and step > " + data.getStep();
    int mstep = adminExecuteQueryNum(sql);
    int instep = 0;
    if(mstep > 0) {
        sql = "update " + boardName + " set step = step + 1 where ref = ";
        sql += data.getRef() + " and step >= " + mstep;
        adminExecuteUpdate(sql);
        instep = mstep;
    } else {
        sql = "select max(step) from " + boardName + " where ref = " + data.getRef();
        instep = adminExecuteQueryNum(sql) + 1;
    }
    int maxNum = getMaxNum(boardName) + 1;
    int depth = data.getDepth();
    sql = "Insert into " + boardName;
    sql += " (num, name, subject, content, writeDate, password, ";
    sql += "count, ref, step, depth, childCount)";
    sql += "values(" + maxNum + ", '" + data.getName() + "', '" + data.getSubject() + "', '" + data.getContent() + "', now()";
    sql += " + ", '" + data.getPassword() + "', 0, " + data.getRef() + ", " + instep + ", " + (++depth) + ", 0)";
    adminExecuteUpdate(sql);
    for(int r = depth - 1; r >= 0; r--) {
        sql = "select max(step) from " + boardName + " where ref = " + data.getRef();
        sql += " and depth = " + r + " and step < " + instep;
        int max_Step = adminExecuteQueryNum(sql);
        sql = "update " + boardName + " set childCount = childCount + 1 ";
        sql += "where ref = " + data.getRef() + " and depth = " + r + " and step = " + max_Step;
        adminExecuteUpdate(sql);
    }
    this.setSuccess(true);
}
```

게시판

▶ BoardManager.java (계속)

```
// 페이지를 위한 메소드
public String getPaging(String boardName, int currentPage) {
    this.currentPage = currentPage;
    int recNum = this.getRecNum(boardName);
    this.totalPage = ((recNum - 1) / 10) + 1;
    String boardLink = "boardList.jsp?boardName=" + boardName;
    int startPage;
    int endPage;
    int linkPage;
    String strList = "";
    startPage = ((currentPage - 1) / PAGES_PER_GROUP) * PAGES_PER_GROUP + 1;
    endPage = (((startPage - 1) + PAGES_PER_GROUP) / PAGES_PER_GROUP) * PAGES_PER_GROUP;
    if(totalPage <= endPage) {
        endPage = totalPage;
    }
    if(currentPage > PAGES_PER_GROUP) {
        linkPage = startPage - 1;
        strList += "<a href='" + boardLink + "&page=" + linkPage + "'>[이전]</a>";
    }else{
        strList += "[이전]";
    }
    strList += "..";
    linkPage = startPage;
    while(linkPage <= endPage) {
        if(linkPage == currentPage){
            strList += " " + currentPage + " ";
        } else {
            strList += "<a href='" + boardLink + "&page=" + linkPage + "'>[" + linkPage + "]</a>";
        }
        linkPage++;
    }
    strList += "..";
    if(totalPage > endPage) {
        linkPage = endPage + 1;
        strList += "<a href='" + boardLink + "&page=" + linkPage + "'>[이후]</a>";
    }else{
        strList += "[이후]";
    }
    return strList;
}
```

게시판

▶ BoardManager.java (계속)

```
// 검색을 위한 변수 설정
public void setSearch(String search) {
    this.search = search;
}
public void setText(String text) {
    this.text = text;
}
// 글을 삭제하기 위한 메소드
public String deleteNotice(String boardName, int num) {
    String msg = new String();
    String sql = "select * from " + boardName + " where num=" + num;
    Vector v = adminExecuteQuery(sql);
    BoardData data = (BoardData)v.elementAt(0);
    int ref = data.getRef();
    int step = data.getStep();
    int depth = data.getDepth();
    int childCount = data.getChildCount();
    if(childCount == 0) {
        sql = "delete from " + boardName + " where num =" + num;
        adminExecuteUpdate(sql);
        for(int r = depth - 1; r >= 0; r--) {
            sql = "select max(Step) from " + boardName + " where ref = " + ref ;
            sql += " and depth = " + r + " and step < " + step;
            int max_Step = adminExecuteQueryNum(sql);
            sql = "update " + boardName + " set childCount = childCount-1 ";
            sql += "where ref = " + ref + " and depth = " + r + " and step =" + max_Step;
            adminExecuteUpdate(sql);
        }
        msg = "OK";
        this.setSuccess(true);
    } else {
        msg = "답변이 있는 글은 삭제할 수 없습니다";
        this.setSuccess(false);
    }
    return msg;
}
```

게시판

▶ BoardManager.java (끝)

```
// 글의 조회수를 증가시키기 위한 메소드
public void increaseCnt(String boardName, int num) {
    String sql = "update " + boardName;
    sql += " set count = count + 1 where num=" + num;
    adminExecuteUpdate(sql);
}

// 이전 글과 다음 글의 번호를 구하기 위한 메소드
public int[] getNextPrevNum(String boardName, int num) {
    int[] nums = new int[2];
    String sql = "select num from " + boardName;
    sql += " where num > " + num + " order by Num desc";
    nums[0] = adminExecuteQueryNum(sql);
    sql = "select num from " + boardName + " where num < " + num;
    nums[1] = adminExecuteQueryNum(sql);
    return nums;
}

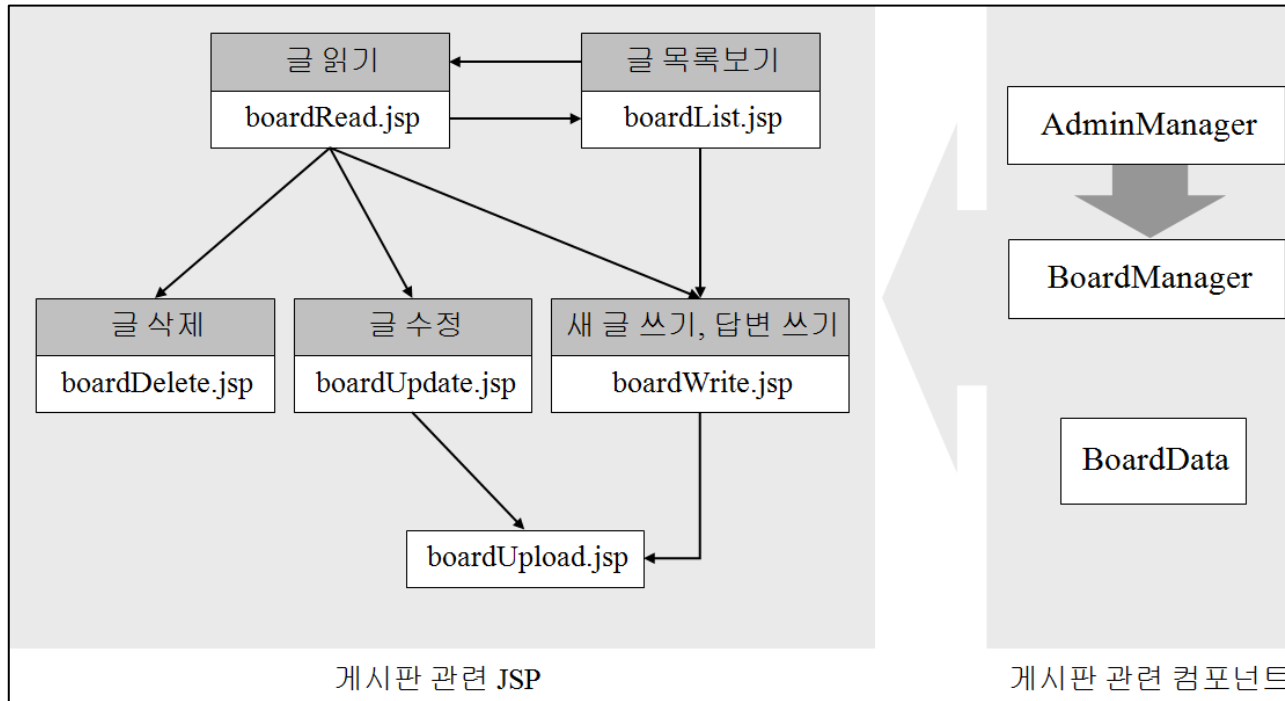
// 작업 수행의 결과를 리턴
public boolean getSuccess(){
    return success;
}

// 작업 수행의 결과를 설정
public void setSuccess(boolean success){
    this.success = success;
}

// 레코드 수를 반환하기 위한 메소드
public int getRecNum(String boardName) {
    String sql = null;
    if(!text.trim().equals("")){
        sql = "select count(*) from " + boardName;
        sql += " where upper(" + search + ") like upper('%" + text + "%)";
    } else {
        sql = "select count(*) from " + boardName;
    }
    this.recNum = adminExecuteQueryNum(sql);
    return this.recNum;
}
}
```


JSP Page

▶ 구조



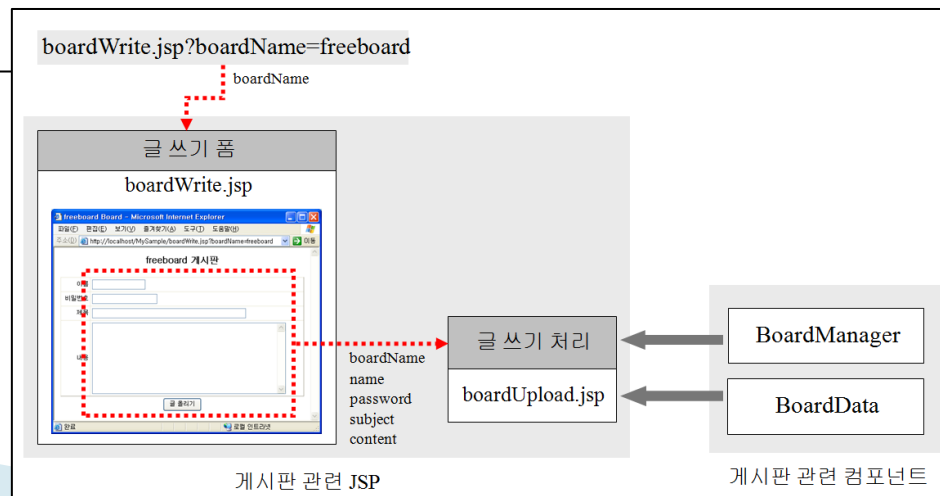
JSP Page

- ▶ 글 쓰기
 - boardWrite.jsp

```
<%@page contentType="text/html; charset=euc-kr"%>
<%@page import="java.util.*"%>
<%String boardName = request.getParameter("boardName");%>
<html>
<head>
<title><%=boardName%> Board</title>
</head>
<body>
<h3><%=boardName%> 게시판</h3>
<form name="write" method="post" action="boardUpload_old.jsp?boardName=<%=boardName%>">
  이름 : <input type="text" name="name" size="16"><br>
  비밀번호 : <input type="password" name="password"><br>
  제목 : <input type="text" name="subject" size="50"><br>
  내용 : <textarea name="content" cols="50" rows="10" scroll="auto"></textarea><br>
  <input type="submit" value="글 올리기">
</form>
</body>
</html>
```

freeboard 게시판

| | |
|--------------------------------------|--------------------------|
| 이름 | <input type="text"/> |
| 비밀번호 | <input type="password"/> |
| 제목 | <input type="text"/> |
| 내용 | <div></div> |
| <input type="button" value="글 올리기"/> | |



JSP Page

- ▶ 글쓰기 요청 처리
 - boardUpload.jsp

```
<%@page contentType="text/html; charset=euc-kr"%>
<%@page import="java.util.*"%>
<jsp:useBean id="boardManager" class="?????.?????.BoardManager"/>
<jsp:useBean id="boardData" class="?????.?????.BoardData"/>
<%
    int currentPage = 0;
    int ref          = 0;
    int step         = 0;
    int depth        = 0;
    int num          = 0;

    String boardName = request.getParameter("boardName");
    String name       = request.getParameter("name");
    String subject    = request.getParameter("subject");
    String content     = request.getParameter("content");
    String password   = request.getParameter("password");

    boardData.setNum(num);
    boardData.setName(name);
    boardData.setSubject(subject);
    boardData.setContent(content);
    boardData.setPassword(password);
    boardData.setRef(ref);
    boardData.setStep(step);
    boardData.setDepth(depth);
    boardManager.writeNotice(boardName, boardData);
    if(boardManager.getSuccess()) {
        boardManager.setSuccess(false);
        response.sendRedirect("boardList.jsp?boardName=" + boardName + "&page=1&search=&text=");
    }
%>
```

JSP Page

▶ **기능 목록**

- Paging

- 글 목록 출력 시 한 Page에 일정한 개수씩 보여줌
- Page 설정

```
int currentPage = 1;
if (request.getParameter("page") != null)
    currentPage = Integer.parseInt(request.getParameter("page"));
```

- 글 목록 Paging

```
Vector v = boardManager.getPage(boardName, currentPage);
String paging = boardManager.getPaging(boardName, currentPage);
```

- 글 목록 출력

```
for(int i = 0; i < v.size(); i++) {
    boardData = (BoardData)v.elementAt(i);
    int depth = boardData.getDepth();
    while(depth > 0) {
        depth--;
        out.print("    ");
    }
    out.print(boardData.getSubject());
}
```

- Paging 출력

<%=paging%>

| 번호 | 제목 | 작성자 | 작성일 | 조회수 |
|----|---|--------|------------|-----|
| 11 | 책목이에게~ | 변은혜 | 2007-03-07 | 0 |
| 9 | [우리 애가 가졌어요..²⁰⁰⁷] | 이명진 | 2007-03-07 | 1 |
| 10 | [RE] 축하드립니다. | jabook | 2007-03-07 | 0 |
| 8 | ㅠㅠㅠㅠ | jabook | 2007-02-26 | 0 |
| 1 | JSP를 잘 하려면? | 이명진 | 2007-02-26 | 8 |
| 2 | [RE] 답변 1 | 변은혜 | 2007-02-26 | 2 |
| 3 | [RE] [RE] 답변 1-1 | jabook | 2007-02-26 | 3 |
| 6 | [RE] [RE] [RE] 답변 1-1-1 | jabook | 2007-02-26 | 0 |
| 7 | [RE] [RE] [RE] 답변 1-1-2 | jabook | 2007-02-26 | 0 |
| 5 | [RE] [RE] 답변 1-2 | jabook | 2007-02-26 | 0 |

[미전].. 1 [2]..[미후]

JSP Page

▶ 글 목록

◦ 검색

```
<form method = post action="boardList.jsp?boardName=<%=boardName%>" name="ser">
  <select name="search" onChange="" align="center">
    <option selected value="subject">제목</option>
    <option value="content">내용</option>
    <option value="name">작성자</option>
  </select>
  <input type="text" name="text">
  <input type="submit" name="search" value="검색">
</form>
```

제목 ▼ [text input] [검색]

제목
내용
작성자

```
if(request.getParameter("text") != null ) {
    search = (String)request.getParameter("search");
    text = request.getParameter("text");
}
boardManager.setSearch(search);
boardManager.setText(text);
Vector v = boardManager.getPage(boardName, currentPage);
```

JSP Page

- ▶ 글 목록
 - boardList.jsp (시작)

```
<%@page contentType="text/html; charset=euc-kr"%>
<%@page import="java.util.*"%>
<%
    String boardName = request.getParameter("boardName");
    String search = "";
    String text = "";
    BoardManager boardManager = new BoardManager();
    if(request.getParameter("text") != null) {
        search = (String)request.getParameter("search");
        if(request.getParameter("kor") != null) {
            text = request.getParameter("text");
        }else {
            text = request.getParameter("text");
        }
    }
    boardManager.setSearch(search);
    boardManager.setText(text);
    int currentPage = 1;
    if(request.getParameter("page") != null)
        currentPage = Integer.parseInt(request.getParameter("page"));
    String paging = boardManager.getPaging(boardName, currentPage);
    Vector v = boardManager.getPage(boardName, currentPage);
%>
```

JSP Page

- ▶ 글 목록
 - boardList.jsp (끝)

```
<html><head><title><%=boardName%> Board</title></head><body>
<h3><%=boardName%> 게시판</h3><hr>
<% BoardData boardData = null;
    if(v.size() > 0) {
        int depth = 0;
        for(int i = 0; i < v.size(); i++) {
            boardData = (BoardData)v.elementAt(i);
            번호 : <%=boardData.getNum()%><br>
            <% depth = boardData.getDepth();
            while(depth > 0) {
                depth--;
                out.println("  ");
            }
            제목 : <a href="boardRead.jsp?boardName=<%=boardName%>
                    &page=<%=currentPage%>&num=<%=boardData.getNum()%>
                    &search=<%=search%>&text=<%=text%>"><%=boardData.getSubject()%></a><br>
            작성자 : <%=boardData.getName()%><br>   작성일 : <%=boardData.getDate()%><br>
            조회수 : <%=boardData.getCount()%><br>
            <br>
            <% }
        } else {
            글이 없습니다.<br>
        }
    }
    <p><%=paging%></p>
    <form method = post action="boardList.jsp?boardName=<%=boardName%>" name="ser">
        <input type="hidden" name="kor" value="ok">
        <select name="search" onChange="" align="center">
            <option selected value="subject">제목</option>
            <option value="content">내용</option>
            <option value="name">작성자</option>
        </select>
        <input type="text" name="text">
        <input type="submit" name="search" value="검색">
    </form>
    <a href="boardList.jsp?boardName=<%=boardName%>">처음목록</a> |
    <a href="boardWrite.jsp?boardName=<%=boardName%>">글쓰기</a>
</body></html>
```

| 번호 | 제목 | 작성자 | 작성일 | 조회수 |
|----|-------------|------|------------|-----|
| 1 | JSP를 잘 하려면? | test | 2007-02-26 | 3 |

[이전].. 1 ..[이후]

제목

[처음목록](#) | [글쓰기](#)

JSP Page

- ▶ 글 읽기

- 글 제목에 글 읽기 Link 작성

```
<a href="boardRead.jsp?page=<%=currentPage%>
    &boardName=<%=boardName%>
    &num=<%=boardData.getNum()%>
    &search=<%=search%>
    &text=<%=text%>" >
    <%=boardData.getSubject()%>
</a>
```

- 글 읽기

```
String currentPage = request.getParameter("page");
String boardName = request.getParameter("boardName");
int num = Integer.parseInt(request.getParameter("num"));
BoardData boardData = boardManager.getNotice(boardName, num);
boardManager.increaseCnt(boardName, num);
```

- 이전 글, 다음 글 번호 출력

```
int[] MaxMin = boardManager.getNextPrevNum(boardName, num);
```

- 다른 Link

```
<form name="replyForm" method="post" action="boardWrite.jsp?boardName=<%=boardName%>" >
... 글 내용 출력 </form>
<a href="javascript:document.replyForm.submit()">답변</a> |
<a href="boardUpdate.jsp?boardName=<%=boardName%>
    &num=<%=boardData.getNum()%>
    &page=<%=currentPage%>">수정</a> |
<a href="boardDelete.jsp?boardName=<%=boardName%>
    &num=<%=boardData.getNum()%>
    &page=<%=currentPage%>">삭제</a> |
<a href="boardList.jsp?boardName=<%=boardName%>
    &page=<%=currentPage%>
    &search=<%=search%>
    &text=<%=text%>">목록</a> |
<a href="boardWrite.jsp?boardName=<%=boardName%>">글쓰기</a>
```


JSP Page

- ▶ 글 읽기
 - boardRead.jsp (시작)

```
<%@page contentType="text/html; charset=euc-kr"%>
<%@page import="java.util.*"%>
<jsp:useBean id="boardManager" class="?????.?????.BoardManager"/>
<%
    String search = "";
    String text = "";
    String currentPage = request.getParameter("page");
    String boardName = request.getParameter("boardName");
    int num = Integer.parseInt(request.getParameter("num"));
    if(request.getParameter("text") != null) {
        search = request.getParameter("search");
        text = request.getParameter("text");
    }
    BoardData boardData = boardManager.getNotice(boardName, num);
    int[] MaxMin = boardManager.getNextPrevNum(boardName, num);
    BoardData pre_data;
    BoardData next_data;
    if(MaxMin[0] <= 0 && MaxMin[1] <= 0) {
        pre_data = null;
        next_data = null;
    } else if(MaxMin[0] <= 0) {
        pre_data = boardManager.getNotice(boardName, MaxMin[1]);
        next_data = null;
    } else if(MaxMin[1] <= 0) {
        pre_data = null;
        next_data = boardManager.getNotice(boardName, MaxMin[0]);
    } else {
        pre_data = boardManager.getNotice(boardName, MaxMin[1]);
        next_data = boardManager.getNotice(boardName, MaxMin[0]);
    }
    boardManager.increaseCnt(boardName, num);
%>
```

freeboard 게시판

| | |
|-------------|--------------------------|
| 글번호 | 1 |
| 제목 | JSP를 잘 하려면? |
| 이름 | test |
| 조회수 | 2 |
| 작성일 | 2007-02-26 |
| 내용 | JSP를 잘 하려면 무엇을 공부해야 하나요? |
| 이전 글이 없습니다. | |
| 다음 글이 없습니다. | |

[답변](#) | [수정](#) | [삭제](#) | [목록](#) | [글쓰기](#)

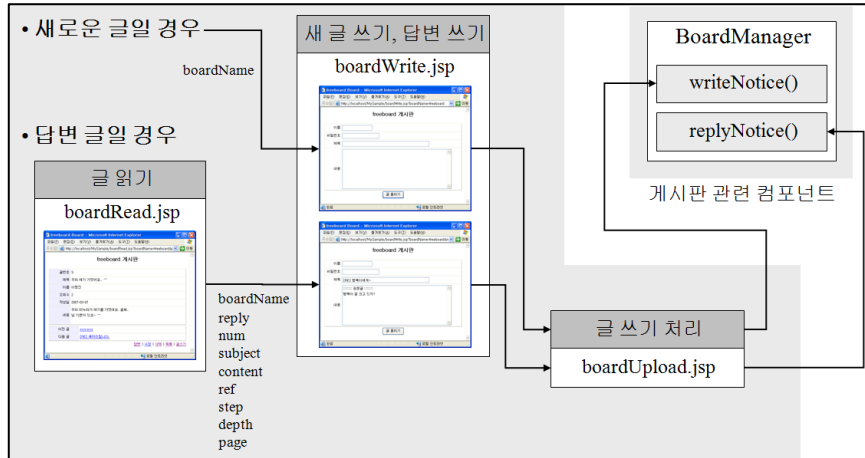
JSP Page

- ▶ 글 읽기
 - boardRead.jsp (끝)

```
<html><head><title><%=boardName%> Board</title></head><body>
<h3><%=boardName%> 게시판</h3><hr>
  <form name="replyForm" method="post" action="boardWrite.jsp?boardName=<%=boardName%>" >
    <input type="hidden" name="subject" value="<%=boardData.getSubject()%>">
    <input type="hidden" name="content" value="<%=boardData.getContent()%>">
    <input type="hidden" name="num" value="<%=boardData.getNum()%>">
    <input type="hidden" name="page" value="<%=currentPage%>">
    <input type="hidden" name="reply" value="ok">
    <input type="hidden" name="ref" value="<%=boardData.getRef()%>">
    <input type="hidden" name="step" value="<%=boardData.getStep()%>">
    <input type="hidden" name="depth" value="<%=boardData.getDepth()%>">
    글번호 : <%=num%><br>
    제목 : <%=boardData.getSubject()%> <br> 작성자 : <%=boardData.getName()%><br>
    조회수 : <%=boardData.getCount()%> <br> 작성일 : <%=boardData.getDate()%><br>
    <pre><%=boardData.getContent()%></pre><br>
    <% if (MaxMin[1] != 0) { %>
      이전 글 :
      <a href="boardRead.jsp?boardName=<%=boardName%>&num=<%=MaxMin[1]%>
        &page=<%=currentPage%>"><%=pre_data.getSubject()%></a><br>
    <% } else { %>
      이전 글이 없습니다.<br>
    <% } if (MaxMin[0] != 0) { %>
      다음 글 :
      <a href="boardRead.jsp?boardName=<%=boardName%>&num=<%=MaxMin[0]%>
        &page=<%=currentPage%>"><%=next_data.getSubject()%></a><br>
    <% } else { %>
      다음 글이 없습니다.<br>
    <% } %>
    <hr><a href="javascript:document.replyForm.submit()">답변</a> |
    <a href="boardUpdate.jsp?boardName=<%=boardName%>&num=<%=boardData.getNum()%>
      &page=<%=currentPage%>">수정</a> |
    <a href="boardDelete.jsp?boardName=<%=boardName%>&num=<%=boardData.getNum()%>
      &page=<%=currentPage%>">삭제</a> |
    <a href="boardList.jsp?boardName=<%=boardName%>&page=<%=currentPage%>
      &search=<%=search%>&text=<%=text%>">목록</a> |
    <a href="boardWrite.jsp?boardName=<%=boardName%>">글쓰기</a>
  </form>
</body></html>
```

JSP Page

▶ 답변 글 쓰기



JSP Page

▶ 답변 글 쓰기

◦ boardRead.jsp 수정

- 답변 글을 위한 Hidden Text Field 작성

```
<input type="hidden" name="reply" value="ok">  
<input type="hidden" name="step" value="<%=boardData.getStep()%>">  
... 숨겨진 텍스트 필드 ...
```

- 답변 글로 전달된 Parameter 처리

```
if(request.getParameter("reply") != null && request.getParameter("reply").equals("ok")) {  
    num      = Integer.parseInt(request.getParameter("num"));  
    reply    = request.getParameter("reply");  
    ref      = request.getParameter("ref");  
    step     = request.getParameter("step");  
    depth    = request.getParameter("depth");  
    subject  = "[Re] " + request.getParameter("subject");  
    content  = request.getParameter("content");  
    contentText = "::::::::: 원문글 ::::::::::\n";  
}
```

JSP Page

▶ 답변 글 쓰기

- boardRead.jsp 수정
- 숨겨진 Text Field 추가

```
<input type="hidden" name="page" value="<%=currentPage%>">
<input type="hidden" name="num" value="<%=num%>">
<input type="hidden" name="ref" value="<%=ref%>">
<input type="hidden" name="step" value="<%=step%>">
<input type="hidden" name="depth" value="<%=depth%>">
<input type="hidden" name="reply" value="<%=reply%>">
```

- 글 제목, 글 내용 Input Box 수정

```
<input type="text" name="subject" size="50" value="<%=subject%>">
<textarea name="content" cols="62" rows="10" scroll="auto">
    <%out.print (contentText) ;out.print (content) ;%>
</textarea>
```

JSP Page

- ▶ 답변 글 쓰기
 - boardRead.jsp (시작)

```
<%@page contentType="text/html; charset=euc-kr"%>
<%@page import="java.util.*"%>
<%
    String boardName = request.getParameter("boardName");
    String currentPage = request.getParameter("page");
    String reply = "no";
    String ref = "";
    String step = "";
    String depth = "";
    String childCount = "";
    String subject="";
    String content = "";
    String contentText = "";
    int num=0;
    BoardData data = null;
    if(request.getParameter("reply") != null && request.getParameter("reply").equals("ok"))
        num = Integer.parseInt(request.getParameter("num"));
        reply = request.getParameter("reply");
        ref = request.getParameter("ref");
        step = request.getParameter("step");
        depth = request.getParameter("depth");
        subject = "[Re] " + request.getParameter("subject");
        content = request.getParameter("content");
        contentText = ":::::::: 원문글 ::::::::::\n";
    }
%>
```

JSP Page

- ▶ 답변 글 쓰기
 - boardRead.jsp (끝)

```
<html><head><title><%=boardName%> Board</title></head><body>
<h3><%=boardName%> 게시판</h3>
<form name="write" method="post" action="boardUpload.jsp?boardName=<%=boardName%>">
  <input type="hidden" name="page" value="<%=currentPage%>">
  <input type="hidden" name="num" value="<%=num%>">
  <input type="hidden" name="ref" value="<%=ref%>">
  <input type="hidden" name="step" value="<%=step%>">
  <input type="hidden" name="depth" value="<%=depth%>">
  <input type="hidden" name="reply" value="<%=reply%>">
  이름 : <input type="text" name="name" size="16"><br>
  비밀번호 : <input type="password" name="password"><br>
  제목 : <input type="text" name="subject" size="50" value="<%=subject%>"><br>
  내용 : <textarea name="content" cols="62" rows="10" scroll="auto">
    <%out.print(contentText);out.print(content);%></textarea><br>
  <input type="submit" value="글 올리기">
</form>
</body></html>
```

JSP Page

▶ 답변 글 쓰기

- reply Parameter에 'ok' 값이 전달되면
 - 답변 글이므로 replyNotice() Method
- 아니면
 - 답변 글이 아니므로 writeNotice() Method
- 답변 글일 경우 Parameter 값 획득

```
if(reply.equals("ok")){  
    ref = Integer.parseInt(request.getParameter("ref"));  
    step = Integer.parseInt(request.getParameter("step"));  
    depth = Integer.parseInt(request.getParameter("depth"));  
}
```

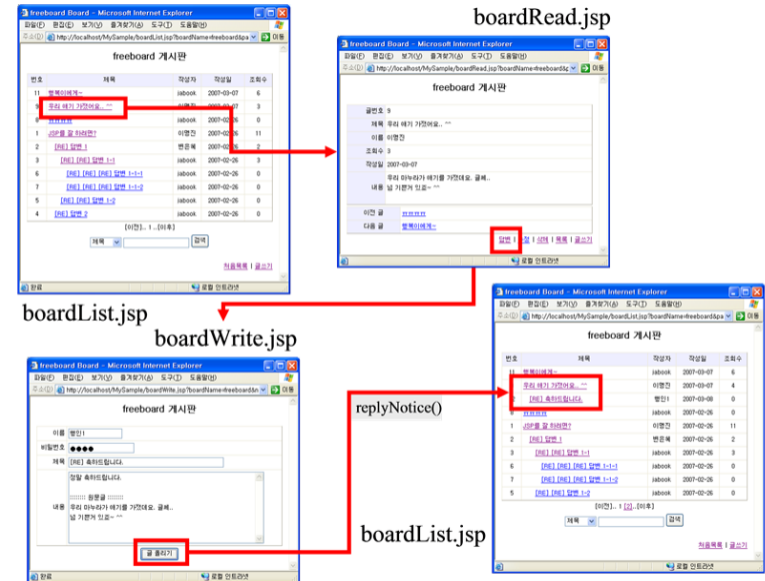
- 새로운 글, 답변 글 여부에 따라 다른 Method 호출

```
if(reply.equals("ok")){  
    boardManager.replyNotice(boardName, boardData);  
} else {  
    boardManager.writeNotice(boardName, boardData);  
} if(boardManager.getSuccess()) {  
    boardManager.setSuccess(false);  
    response.sendRedirect("boardList.jsp?boardName=" + boardName + "&page=" + currentPage);  
}
```


JSP Page

- ▶ 답변 글 쓰기
 - boardUpload.jsp

```
<%@page contentType="text/html; charset=euc-kr"%>
<%@page import="java.util.*"%>
<jsp:useBean id="boardManager" class="?????.?????.BoardManager"/>
<jsp:useBean id="boardData" class="?????.?????.BoardData"/>
<%
    int currentPage = 0;    int ref = 0;    int step = 0;
    int depth = 0;        int num = 0;
    String boardName = request.getParameter("boardName");
    String name = request.getParameter("name");
    String subject = request.getParameter("subject");
    String content = request.getParameter("content");
    String password = request.getParameter("password");
    String reply = request.getParameter("reply");
    if(reply.equals("ok")){
        currentPage = Integer.parseInt(request.getParameter("page"));
        ref = Integer.parseInt(request.getParameter("ref"));
        step = Integer.parseInt(request.getParameter("step"));
        depth = Integer.parseInt(request.getParameter("depth"));
    }
    boardData.setNum(num);        boardData.setName(name);
    boardData.setSubject(subject); boardData.setContent(content);
    boardData.setPassword(password); boardData.setRef(ref);
    boardData.setStep(step);      boardData.setDepth(depth);
    if(reply.equals("ok")){
        boardManager.replyNotice(boardName, boardData);
        if(boardManager.getSuccess()){
            boardManager.setSuccess(false);
            response.sendRedirect("boardList.jsp?boardName=" + boardName + "&page=" + currentPage);
        }
    }
    else {
        boardManager.writeNotice(boardName, boardData);
        if(boardManager.getSuccess()){
            boardManager.setSuccess(false);
            response.sendRedirect("boardList.jsp?boardName=" + boardName + "&page=1&search=&text=");
        }
    }
}
```



JSP Page

▶ 글 수정

- 수정 가능한 항목이 정해져 있음 (제목, 이름, 내용 등)
- 글의 비밀번호 비교를 통해 수정 권한 검사

```
String flag = "false";
flag = (request.getParameter("flag") == null) ? "false" : request.getParameter("flag");

String password = new String();
password = request.getParameter("password");
String noticePassword = boardManager.getPassword(boardName, num);
if(password != null && (password.trim()).equals(noticePassword))
    boardManager.setSuccess(true);
```

- 글 수정 권한 확인 후 입력 양식 출력

```
<% if(boardManager.getSuccess()) {
    ... 글 수정 양식 출력 ...
} else { %>
    <form name="chkPass"
        method="post"
        action="boardUpdate.jsp?boardName=<%=boardName%>
            &num=<%=num%>
            &page=<%=currentPage%>">

        <input type="password" name="password">
        <input type="submit" value="확인">
    </form>
<% } %>
```

```
if(boardManager.getSuccess()) {
    boardManager.setSuccess(false);
    ... 수정 양식 출력 ...
} else {
    ... 패스워드 입력 양식 출력 ...
}
```

JSP Page

- ▶ 글 수정
 - 글 수정 양식

```
<form name="write" method="post" action="boardUpdate.jsp?boardName=<%=boardName%>">
  <input type="hidden" name="flag" value="true">
  <input type="hidden" name="num" value="<%=boardData.getNum()%>">
  <input type="hidden" name="page" value="<%=currentPage%>">
  <input type="text" name="name" size="16" value="<%=boardData.getName()%>">
  <input type="password" name="password" value="<%=boardData.getPassword()%>">
  <input type="text" name="subject" size="50" value="<%=boardData.getSubject()%>">
  <textarea name="content" cols="62" rows="10" scroll="auto">
    <%=boardData.getContent()%>
  </textarea>
  <input type="submit" value="수정 하기">
</form>
```

- 글 수정 후 Data 셋팅

```
editData.setNum(num);
editData.setName(request.getParameter("name"));
editData.setSubject(request.getParameter("subject"));
editData.setContent(request.getParameter("content"));
```

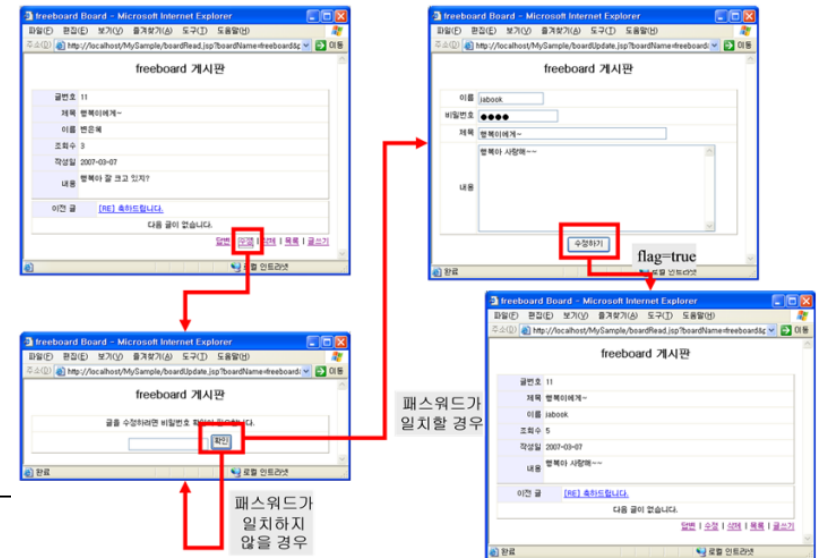
- 수정을 위해 Method 호출

```
if(flag.equals("true")){
  boardManager.updateNotice(boardName, editData);
  if(boardManager.getSuccess()) {
    boardManager.setSuccess(false);
    response.sendRedirect("boardList.jsp?boardName=" + boardName + "&page=" + currentPage);
  } else {
    out.println("<script>alert('글이 수정 되지않았습니다');</script>");
  }
}
```

JSP Page

- ▶ 글 수정
 - boardUpdate.jsp (시작)

```
<%@page contentType="text/html; charset=euc-kr"%>
<%@page import="java.util.*"%>
<jsp:useBean id="boardManager" class="?????.?????.BoardManager"/>
<jsp:useBean id="editData" class="?????.?????.BoardData"/>
<%
    String boardName = (String)request.getParameter("boardName");
    int num = Integer.parseInt((String)request.getParameter("num"));
    String currentPage = (String)request.getParameter("page");
    String flag = "false";
    flag = (request.getParameter("flag") == null) ? "false" : request.getParameter("flag");
    if(flag.equals("true")) {
        boardManager.setSuccess(true);
    }
    String password = new String();
    password = request.getParameter("password");
    String noticePassword = boardManager.getPassword(boardName, num);
    if(password != null && (password.trim()).equals(noticePassword))
        boardManager.setSuccess(true);
%>
```



JSP Page

- ▶ 글 수정
 - boardUpdate.jsp (끝)

```
<html><head><title><%=boardName%> Board</title></head><body>
<h3><%=boardName%> 게시판</h3><hr>
<% if (boardManager.getSuccess()) {
    boardManager.setSuccess(false);
    BoardData boardData = boardManager.getNotice(boardName, num);
%>

    <form name="write" method="post" action="boardUpdate.jsp?boardName=<%=boardName%>">
        <input type="hidden" name="flag" value="true">
        <input type="hidden" name="num" value="<%=boardData.getNum()%>">
        <input type="hidden" name="page" value="<%=currentPage%>">
        이름 : <input type="text" name="name" size="16" value="<%=boardData.getName()%>"><br>
        비밀번호 : <input type="password" name="password" value="<%=boardData.getPassword()%>"><br>
        제목 : <input type="text" name="subject" size="50" value="<%=boardData.getSubject()%>"><br>
        내용 : <textarea name="content" cols="50" rows="10" scroll="auto"><%=boardData.getContent()%></textarea><br>
        <input type="submit" value="수정하기">
    </form>

<%
    editData.setNum(num);
    editData.setName(request.getParameter("name"));
    editData.setSubject(request.getParameter("subject"));
    editData.setContent(request.getParameter("content"));
    if (flag.equals("true")) {
        boardManager.updateNotice(boardName, editData);
        if (boardManager.getSuccess()) {
            boardManager.setSuccess(false);
            response.sendRedirect("boardList.jsp?boardName=" + boardName + "&page=" + currentPage);
        } else {
            out.println("<script>alert('글이 수정되지않았습니다');</script>");
            response.sendRedirect("boardRead.jsp?boardName=" + boardName + "&page=" + currentPage + "&num=" +
                ((String) request.getParameter("num")));
        }
    }
} else {
%>

    <form name="chkPass" method="post" action="boardUpdate.jsp?boardName=<%=boardName%>&num=<%=num%>&page=<%=currentPage%>">
        글을 수정하려면 비밀번호 확인이 필요합니다.<br>
        <input type="password" name="password"><input type="submit" value="확인">
    </form>

<%
} %>
</body></html>
```

JSP Page

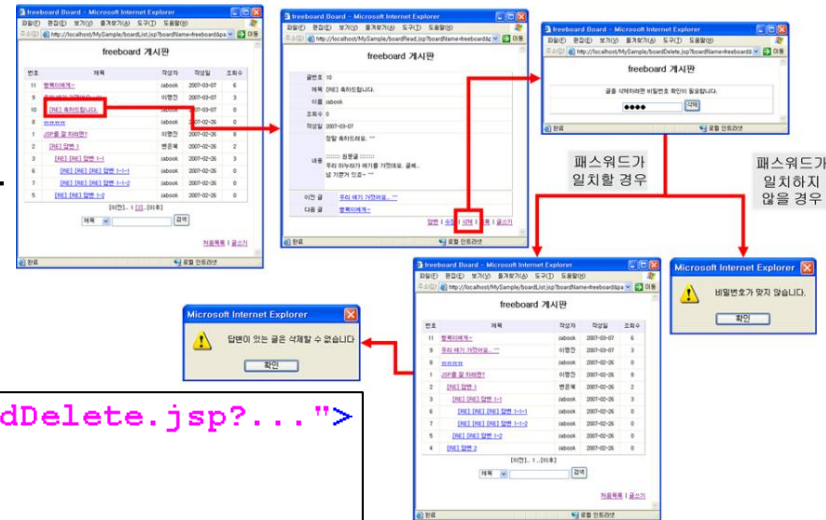
▶ 글 삭제

- 답변 글이 있을 경우 삭제 여부
- 글 삭제 권한 검사
- 비밀번호 입력 양식

```
<form name="chkPass" method="post" action="boardDelete.jsp?...">  
  <input type="password" name="password">  
  <input type="submit" value="삭제">  
</form>
```

◦ 비밀번호 검사 후 글 삭제

```
if(request.getParameter("password") != null) {  
    String noticePassword = boardManager.getPassword(boardName, num);  
    if(!noticePassword.equals(password)) {  
        out.print("비밀 번호가 맞지 않습니다. 다시 입력해 주세요. ");  
    } else {  
        String msg = boardManager.deleteNotice(boardName, num);  
    }  
}
```



JSP Page

- ▶ 글 삭제
 - boardDelete.jsp

```
<%@page contentType="text/html; charset=euc-kr"%>
<%@page import="java.util.*"%>
<jsp:useBean id="boardManager" class="org.jabook.noveljsp.board.BoardManager"/>
<%
String currentPage = request.getParameter("page");
String boardName = request.getParameter("boardName");
int num = Integer.parseInt(request.getParameter("num"));
String password = request.getParameter("password");
%>
<html><head><title><%=boardName%> Board</title></head><body>
<h3><%=boardName%> 게시판</h3><hr>
<form name="chkPass" method="post" action="boardDelete.jsp?boardName=<%=boardName%>&num=<%=num%>&page=<%=currentPage%>">
<%
if(request.getParameter("password") != null) {
    String noticePassword = boardManager.getPassword(boardName, num);
    if(!noticePassword.equals(password)) {
        out.print("<script>alert('비밀번호가 맞지 않습니다.');
```

오늘 숙제

- ▶ 집에서 해보자!
- ▶ 예제파일 분석
 - ▶ 모방은 창조의 어머니