

12. 게시판 구현 - 글 상세보기, 글 수정 구현

content.jsp (: 목록에서 게시글을 클릭 → 글 상세내용 보기)

```
<%
//list.jsp? 링크 content.jsp?num=3&pageNum=1 로 넘어오면
int num = Integer.parseInt(request.getParameter("num"));
String pageNum = request.getParameter("pageNum");
SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm"); // 작성 형식
BoardDAO dbPro = new BoardDAO();
BoardDTO article = dbPro.getArticle(num);
//링크 문자열의 길이를 줄이기 위해서
int ref = article.getRef();
int re_step=article.getRe_step();
int re_level = article.getRe_level();
```

게시글을 클릭하면, list.jsp로부터 게시물 번호 num을
받아서 그 객체에 해당하는 article 정보를 DB로부터 가
져와 article에 넣어줌.

그리고 해당 article의 ref,re_step,re_level을 setting

```
%>
<body bgcolor="#e0ffff">
<b>글내용 보기</b>
<form>
<table width="500" border="1" cellspacing="0" cellpadding="0" bgcolor="#e0ffff" align="center">
  <tr height="30">
    <td align="center" width="125" bgcolor="#b0e0e6">글번호</td>
    <td align="center" width="125" align="center">
      <%=article.getNum() %></td>
    <td align="center" width="125" bgcolor="#b0e0e6">조회수</td>
    <td align="center" width="125" align="center">
      <%=article.getReadcount() %></td>
  </tr>
  <tr>
    <td align="center" colspan="4">중간 생략</td>
  </tr>
  <tr height="30">
    <td align="center" width="125" bgcolor="#b0e0e6">작성자</td>
    <td align="center" width="125" align="center">
      <%=article.getWriter() %></td>
    <td align="center" width="125" bgcolor="#b0e0e6">작성일</td>
    <td align="center" width="125" align="center">
      <%=sdf.format(article.getReg_date()) %></td>
  </tr>
  <tr height="30">
```

각 필드의 값들을 [표현식]으로 넣어줌

링크설정

: 글수정, 글삭제, 답글, 글목록

```
onlick="document.location.href='updateForm.jsp?num=<%=article.getNum()%>&pageNum=<%=pageNum%>' ">  
    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~  
    <input type="button" value="글삭제"  
  
onlick="document.location.href='deleteForm.jsp?num=<%=article.getNum()%>&pageNum=<%=pageNum%>' ">  
    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~  
    <input type="button" value="답글쓰기"
```



```
<br>
```

```
onclick="document.location.href='writeForm.jsp?num=<%=num%>&ref=<%=ref%>&re_step=<%=re_step%>&re_level=
```

<%=pageNum%>' "

```
>" ">  
  
    &nbsp;&nbsp;&nbsp;&nbsp;&~  
    <input type="button" value="글목록"</td>
```

BoardDAO.jsp (: getArticle() method 구현)

//4.글상세보기

//형식) select * from board where num=? =? 반환형 O 매개변수 O

//형식2) update board set readcount = readcount +1 where num=?

```
public BoardDTO getArticle(int num) {
```

```
    BoardDTO article = null; // 한 개의 레코드를 담는 객체
```

```
    try {
```

```
        con = pool.getConnection();
```

```
        sql="update board set readcount=readcount+1 where num=?";
```

```
        pstmt = con.prepareStatement(sql);
```

```
        pstmt.setInt(1, num);
```

```
        int update = pstmt.executeUpdate();
```

```
        System.out.println("조회수 증가유무:"+update);
```

```
        // 2. 조회수를 증가시킨 게시물을 찾아서 출력
```

```
        sql="select * from board where num=?";
```

```
        pstmt = con.prepareStatement(sql);
```

```
        pstmt.setInt(1, num);
```

```
        rs = pstmt.executeQuery();
```

```
        if(rs.next()) { // 만약 레코드가 1개라도 존재한다면
```

```
            // 형식) articleList = new 자식클래스명();
```

```
            article = makeArticleFromResult();
```

```
        }catch(Exception e) {
```

```
            System.out.println("getArticle() method error:"+e);
```

```
        }finally {
```

```
            pool.freeConnection(con,pstmt,rs);
```

```
        }
```

```
        return article;
```

```
    }
```

DB접속 및
SQL 구문 설정

클릭시 조회수 증가

updateForm.jsp (: 글 수정을 눌렀을 때 넘어가는 페이지)

```
<%
    //content.jsp → 글수정 → updateForm.jsp?num=?&pageNum=?
    //파라미터는 content.jsp와 동일하기 때문에 동일한 코드를 사용
    int num = Integer.parseInt(request.getParameter("num"));
    String pageNum = request.getParameter("pageNum");
    BoardDAO dbPro = new BoardDAO();
    BoardDTO article = dbPro.updateGetArticle(num); //
```

조회수 증가 기능 없음

(ref, re_step, re_level도 필요없음, 수정되면 안됨)

// 그냥 getArticle에서는 readcount+1 을 해줌

```
%>
<body bgcolor="#e0ffff">
<b>글수정</b>
<form method="post" name="writeform" action="updatePro.jsp?pageNum=<%=pageNum%>" onsubmit="return writeSave()">
<table width="400" border="1" cellspacing="0" cellpadding="0" bgcolor="#e0ffff" align="center">
```

submit 하면 script.js의 writeSave() 를 실행
하고 updatePro.jsp?~~~ 로 보낸다.

```
    <tr>
        <td width="70" bgcolor="#b0e0e6" align="center">이 름</td>
        <td align="left" width="330">
            <input type="text" size="10" maxlength="10" name="writer" value="<%=article.getWriter()%>">
            <input type="hidden" name="num" value="<%=article.getNum()%>">
        </td>
    </tr>
    <tr>
        <td width="70" bgcolor="#b0e0e6" align="center">제 목</td>
        <td align="left" width="330">
            <input type="text" size="40" maxlength="50" name="subject" value="<%=article.getSubject()%>"> </td>
    </tr>
    <tr>
        <td width="70" bgcolor="#b0e0e6" align="center">Email</td>
        <td align="left" width="330">
            <input type="text" size="40" maxlength="30" name="email" value="<%=article.getEmail()%>"> </td>
    </tr>
    <tr>
        <td width="70" bgcolor="#b0e0e6" align="center">내 용</td>
        <td align="left" width="330">
            <textarea name="content" rows="13" cols="40"><%=article.getContent() %></textarea> </td>
    </tr>
    <tr>
        <td width="70" bgcolor="#b0e0e6" align="center">비밀번호</td>
        <td align="left" width="330">
            <input type="password" size="8" maxlength="12" name="passwd">
```

표시만 되고 수정은 하지 않지만,
넘겨줘야만 하는 값은 hidden 속성으로
꼭! 넘겨줘야만 한다

해당 필드들이 표시되는 input 에
해당 article의 필드들을 [표현식]으로
불러와서 넣어 사용자에게 보여주는 코드

암호 (password는 안보여줌)

```
    </td>
</tr>
<tr>
    <td colspan=2 bgcolor="#b0e0e6" align="center">
        <input type="submit" value="글수정" >
        <input type="reset" value="다시작성">
        <input type="button" value="목록보기" onclick="document.location.href='list.jsp?pageNum=<%=pageNum%>'">
    </td>
</tr>
</table>
</form>
```

updateProc.jsp (: 수정 form 에서 수정을 눌렀을 때 넘어가는 페이지)

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ page import="devtaco.board.*" %>
<%
    request.setCharacterEncoding("utf-8"); // 한글처리
%>
<jsp:useBean id="article" class="devtaco.board.BoardDTO"/>
<jsp:setProperty name="article" property="*" />

<!-- 그 외에 시스템을 통해서 입력받을 값은 아래에 구현 -->
<%
    //정상적으로 수정이 되었을때, 수정이 된 레코드(블록)로 이동 → 게시물이 수정된 페이지로 이동하라
    String pageNum= request.getParameter("pageNum");
    BoardDAO dbPro = new BoardDAO(); //updateArticle() 호출을 위한 객체
    int check = dbPro.updateArticle(article);

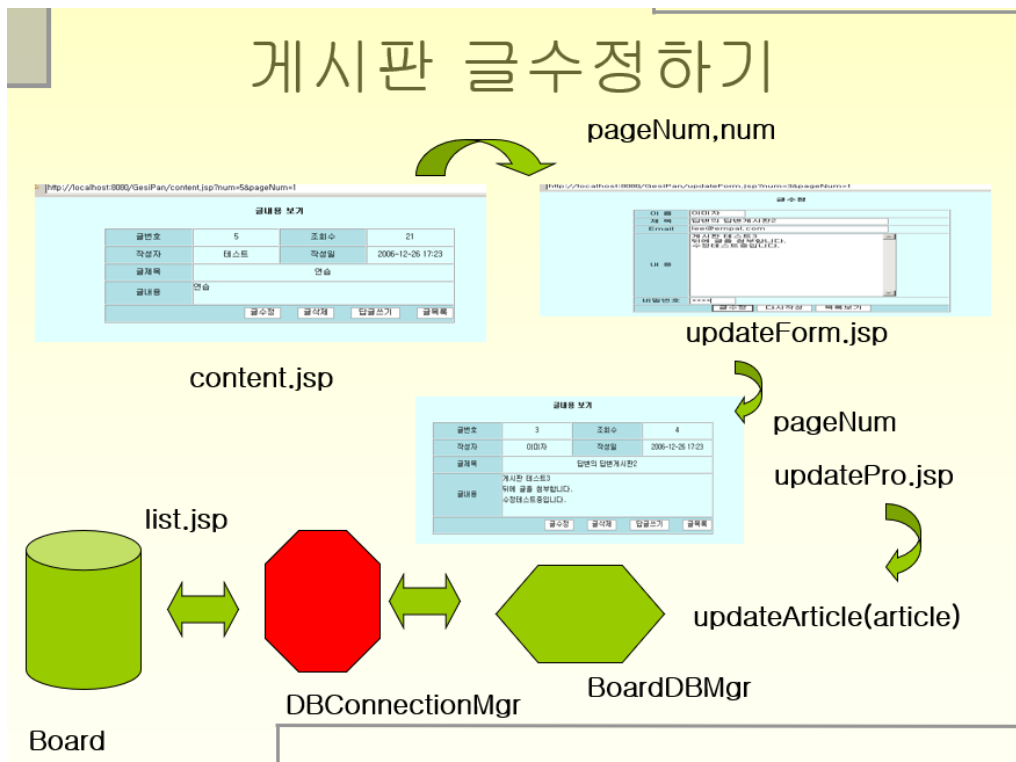
    if(check==1){ //수정이 성공했다면
        //response.sendRedirect("list.jsp");
        //http-equiv="Refresh" ==> 새로고침 옵션 content="초단위(몇초동안 멈춘후); url="이동페이지명";
    }

    <meta http-equiv="Refresh" content="0;url=list.jsp?pageNum=<%=pageNum%>" />
%>
<% } else{ %>
<script>
    alert("비밀번호가 맞지 않습니다. 다시 비밀번호를 확인해 주십시오.");
    history.go(-1);
</script>
<% } %>
```

useBean 태그를 사용해서 article 객체 생성
property는 * 모든 값

updateArticle() method 호출

수정 된 게시물이 있는 페이지로 이동



BoardDAO.jsp (: updateGetArticle(int num)글 수정을 눌렀을 때 해당 글의 data 를 가져오는 method)

//5.수정할 글 가져오기

//select * from board where num=? > 조회수는 증가되지 않아야함

public BoardDTO updateGetArticle(int num){

BoardDTO article = null;

try {

con=pool.getConnection();

sql="select * from board where num=?";

pstmt = con.prepareStatement(sql);

pstmt.setInt(1, num);

rs = pstmt.executeQuery();

if(rs.next()) {

article = makeArticleFromResult(); // 중복코딩 배제

}

}catch(Exception e) {

System.out.println("updateGetArticle() method error:"+e);

}finally {

pool.freeConnection(con,pstmt,rs);

}

return article;

}

DB 연결 및
SQL 구문 설정

BoardDAO.jsp (: updateGetArticle(int num)글 수정을 눌렀을 때 해당 글의 data 를 가져오는 method)

```
// 6. 글 수정하는 method 호출 -> insertArticle과 거의 동일 -> 1) SQL 2) 암호를 인증
public int updateArticle(BoardDTO article) { // updatePro.jsp 에서 호출(암호를 확인)
    String dbpasswd = null; // db에서 찾은 암호 저장
    int x = -1; // 게시물의 수정성공 유무 (초기화)
    // 1.암호를 먼저 찾아서 비교 -> 승낙 -> 2)update 수행이 가능

    try {
        con = pool.getConnection();
        sql = "select passwd from board where num=?"; // 최대값 +1 -> 실제 저장할 게시물 번호를 생성
        pstmt = con.prepareStatement(sql);
        pstmt.setInt(1, article.getNum());
        rs = pstmt.executeQuery();
        if (rs.next()) { // 데이터가 존재한다면
            dbpasswd = rs.getString("passwd");
            if (dbpasswd.contentEquals(article.getPasswd())) {
                // 수정가능한 필드를 수정해서 넣어줌
                sql = "update board set writer=?,email=?,subject=?,passwd=?,content=? where num=?";
                pstmt = con.prepareStatement(sql); // 웹에 입력, 저장 -> setter Method 호출
                pstmt.setString(1, article.getWriter());
                pstmt.setString(2, article.getEmail());
                pstmt.setString(3, article.getSubject());
                pstmt.setString(4, article.getPasswd());
                pstmt.setString(5, article.getContent());
                pstmt.setInt(6, article.getNum());
                int update = pstmt.executeUpdate();
                x = 1;
            } else {
                x = 0; // 암호가 틀린경우(수정 실패)
            }
        } // if(rs.next())
    } catch (Exception e) {
        System.out.println("updateArticle method오류 : " + e);
    } finally {
        pool.freeConnection(con,pstmt,rs);
    }
    return x;
}
```

DB의 암호와 웹상에 입력한 암호가
같은지 체크

수정된 값들을
가져온 article에 set 해줌