# 회원제 게시판 만들기 (게시판 부분)

### LoginService.java 변경사항(페이지 처리를 위해)

|  |
| --- |
| public class LoginService implements Service{    @Override  public ForwardService execute(HttpServletRequest req, HttpServletResponse res)  throws Exception {  MemberDao mDao = MemberDao.getInstance();  ForwardService forward = new ForwardService();    HttpSession session = req.getSession();    String id = req.getParameter("id");  String pw = req.getParameter("pw");    System.out.println("id: " + id + ", pw: " + pw);    String name = mDao.loginProc(id, pw);    if(name != null) {  System.out.println("name: " + name);  session.setAttribute("id", id);  session.setAttribute("name", name);  forward.setRedirect(true);  **forward.setPath("./list.bo?pnum=1");**  }  else {  session.setAttribute("msg", "0");  forward.setRedirect(true);  forward.setPath("./");  }    return forward;  }  } |

### Board.java

|  |
| --- |
| package com.bean;  public class Board {  private String bNo;  private String userId;  private String title;  private String content;  private String bDate;  private String fileName;    public String getbNo() {  return bNo;  }  public void setbNo(String bNo) {  this.bNo = bNo;  }  public String getUserId() {  return userId;  }  public void setUserId(String userId) {  this.userId = userId;  }  public String getTitle() {  return title;  }  public void setTitle(String title) {  this.title = title;  }  public String getContent() {  return content;  }  public void setContent(String content) {  this.content = content;  }  public String getbDate() {  return bDate;  }  public void setbDate(String bDate) {  this.bDate = bDate;  }  public String getFileName() {  return fileName;  }  public void setFileName(String fileName) {  this.fileName = fileName;  }  } |

## 목록 보기 처리

### BoardController.java

|  |
| --- |
| public class BoardController extends HttpServlet {  …  private void doProcess(HttpServletRequest request, HttpServletResponse response)  throws ServletException, IOException {  String command = request.getRequestURI();  HttpSession session = request.getSession();    Service service = null;  ForwardService forward = null;    if (command.indexOf(request.getContextPath()) == 0) {  command = command.substring(request.getContextPath().length());  }    try {  if(command.equals("/list.bo")) {  service = new ListService();  forward = service.execute(request, response);  }    if(forward != null){  if (forward.isRedirect()) {  response.sendRedirect(forward.getPath());  } else {  RequestDispatcher dispatcher = request  .getRequestDispatcher(forward.getPath());  dispatcher.forward(request, response);  }  }  } catch (Exception e) {  // TODO Auto-generated catch block  e.printStackTrace();  }    }  } |

### ListService.java – 패키지 : com.service.board

|  |
| --- |
| public class ListService implements Service {  @Override  public ForwardService execute(HttpServletRequest req, HttpServletResponse res)  throws Exception {  ForwardService forward = new ForwardService();  HttpSession session = req.getSession();  String userId = (String)session.getAttribute("id");    int pageNum = 1;  String pNum = req.getParameter("pnum");  if(pNum != null) {  pageNum = Integer.parseInt(req.getParameter("pnum"));  }    BoardDao bDao = BoardDao.getInstance();    int listCnt = bDao.getBoardListCnt();//전체 글 개수 구하기  ArrayList<Board> bList = bDao.getBoardList(pageNum);    int maxPage = (int)(listCnt/10.0 + 0.9); // 전체 페이지 수  int startPage = (int)(pageNum/5.0 + 0.8) \* 5 - 4; //시작 페이지 번호  int endPage = startPage + 4; //마지막 페이지 번호  if(endPage > maxPage) {  endPage = maxPage;  }  // 4개 페이지번호 저장  req.setAttribute("spage", pageNum);  req.setAttribute("maxPage", maxPage);  req.setAttribute("startPage", startPage);  req.setAttribute("endPage", endPage);  req.setAttribute("bList", bList);    forward.setRedirect(false);  forward.setPath("/board/boardList.jsp");    return forward;  }  } |

### BoardDao.java – 패키지 : com.dao

|  |
| --- |
| public class BoardDao {  private static BoardDao *instance*;  private BoardDao() { }  public static BoardDao getInstance() {  if(*instance* == null) {  *instance* = new BoardDao();  }  return *instance*;  }  public int getBoardListCnt() {  int result = 0;  String query = "select count(\*) as cnt from boardTbl;";    Connection conn = null;  Statement stmt = null;  ResultSet rs = null;    try {  conn = DBConnection.*getConnection*();  stmt = conn.createStatement();  rs = stmt.executeQuery(query);    if(rs.next()) {  result = rs.getInt("cnt");  }  } catch (ClassNotFoundException | SQLException | NamingException e) {  // TODO Auto-generated catch block  e.printStackTrace();  } finally {  try{  if(rs != null) { rs.close(); }  if(stmt != null){ stmt.close(); }  if(conn != null){ conn.close(); }  }catch(Exception e){  e.printStackTrace();  }  }  return result;  }    public ArrayList<Board> getBoardList(){  ArrayList<Board> bList = new ArrayList<>();    String query = "select \* from boardTbl order by bNo DESC limit ?, 10;";    Connection conn = null;  PreparedStatement pstmt = null;  ResultSet rs = null;  try {  conn = DBConnection.*getConnection*();  pstmt = conn.createStatement();  pstmt.setInt(1, limitNum);  rs = pstmt.executeQuery();    while(rs.next()) {  Board board = new Board();  board.setbNo(String.*valueOf*(rs.getInt("bNo")));  board.setUserId(rs.getString("userId"));  board.setTitle(rs.getString("title"));  board.setContent(rs.getString("content"));  board.setbDate(rs.getString("bDate"));  bList.add(board);  }  } catch (ClassNotFoundException | SQLException | NamingException e) {  e.printStackTrace();  throw new RuntimeException(e.getMessage());  } finally {  try{  if(rs != null) { rs.close(); }  if(pstmt != null){ pstmt.close(); }  if(conn != null){ conn.close(); }  }catch(Exception e){  e.printStackTrace();  throw new RuntimeException(e.getMessage());  }  }  return bList;  }  } |

### boardList.jsp

|  |
| --- |
| <%@ page language="java" contentType="text/html; charset=UTF-8"  pageEncoding="UTF-8"%>  <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>  <!DOCTYPE html>  <html>  <head>  <meta charset="UTF-8">  <title>게시판</title>  <script>  function leave(){  var res = confirm('회원을 탈퇴하시겠습니까?');  console.log(res);  if(res){  location.href='./leave.do'  }  }  </script>  <c:if test="${empty id}">  <script>  alert("잘못된 접근입니다.");  location.href="./";  </script>  </c:if>  <style>  th{  background-color: skyblue;  text-align: center  }  </style>  </head>  <body>  <h3>회원제 게시판</h3>  <table>  <tr>  <td width="600" style="text-align: right">${name}님 환영합니다.  <input type="button" value="로그아웃"  onclick="location.href='./logout.do'">  <input type="button" value="회원탈퇴" onclick="leave()">  </td>  </tr>  </table>  <br>  <table>  <tr>  <th width="50">번호</th>  <th width="250">제목</th>  <th width="100">작성자</th>  <th width="180">작성일시</th>  </tr>  <c:forEach var="board" items="${bList}">  <tr>  <td align="center">${board.bNo}</td>  <td align="center"><a href="#">${board.title}</a></td>  <td align="center">${board.userId}</td>  <td align="center">${board.bDate}</td>  </tr>  </c:forEach>  <tr></tr>  <tr>  <td colspan=*"4"* align=*"center"*>  <c:if test=*"*${startPage != 1}*"*>  <a href=*'./list.bo?pnum=*${startPage-1}*'*>[이전]</a>  </c:if>  <c:forEach var=*"pageNum"* begin=*"*${startPage}*"* end=*"*${endPage}*"*>  <c:if test=*"*${pageNum == spage}*"*>  ${pageNum}&nbsp;  </c:if>  <c:if test=*"*${pageNum != spage}*"*>  <a href=*'./list.bo?pnum=*${pageNum}*'*>${pageNum}&nbsp;</a>  </c:if>  </c:forEach>  <c:if test=*"*${endPage != maxPage }*"*>  <a href=*'./list.bo?pnum=*${endPage+1}*'*>[다음]</a>  </c:if>  </td>  </tr>  </table>  <br>  <input type="button" value="글쓰기" onclick="location.href='writeFrm.bo'">  </body>  </html> |

## 글쓰기 처리

### writeFrm.jsp

|  |
| --- |
| <%@ page language="java" contentType="text/html; charset=UTF-8"  pageEncoding="UTF-8"%>  <!DOCTYPE html>  <html>  <head>  <meta charset="UTF-8">  <title>글쓰기</title>  </head>  <h3>게시글 작성</h3>  <body>  <form action="./write.bo" method="post" enctype="multipart/form-data">  <table>  <tr>  <td width="100">아이디</td>  <td><input type="text" name="userId" value="${id}"  readonly="readonly"></td>  </tr>  <tr>  <td>제목</td>  <td><input type="text" name="title" size="60" required="required">  </td>  </tr>  <tr>  <td>내용</td>  <td><textarea name="content" cols="62" rows="20"></textarea></td>  </tr>  <tr>  <td>파일 첨부</td>  <td><input type="file" name="file"></td>  </tr>  <tr>  <td colspan="2" align="center">  <input type="submit" value="저장" />  <input type="button" value="취소"  onclick="location.href='./list.bo?pnum=1'">  </td>  </tr>  </table>    </form>  </body>  </html> |

### BoardController.java

|  |
| --- |
| …  else if(command.equals("/writeFrm.bo")) {  forward = new ForwardService();  forward.setRedirect(false);  forward.setPath("/board/writeFrm.jsp");  }  else if(command.equals("/write.bo")) {  service = new WriteService();  forward = service.execute(request, response);  }  … |

### WriteService.java – 패키지 : com.service.board

|  |
| --- |
| public class WriteService implements Service {  @Override  public ForwardService execute(HttpServletRequest req, HttpServletResponse res)  throws Exception {  ForwardService forward = new ForwardService();  req.setCharacterEncoding("UTF-8");  HttpSession session = req.getSession();  //Board board = new Board();  //파일 업로드 처리  Board board = FileProcess.fileUpload(req);  BoardDao dDao = BoardDao.getInstance();  int result = dDao.writeBoard(board);  if(result > 0) {  session.setAttribute("bmsg", "1");  }  else {  session.setAttribute("bmsg", "2");  }    forward.setRedirect(true);  forward.setPath("./list.bo?pnum=1");  return forward;  }  } |

### FileProcess.java – 패키지 : com.util

|  |
| --- |
| public class FileProcess {  public static Board fileUpload(HttpServletRequest req) {  Board board = new Board();  int fileSize = 5\*1024\*1024;//5MB 제한    String upPath = req.getServletContext().getRealPath("/board/uploadfiles");  System.out.println("upPath: " + upPath);    try {  MultipartRequest multi = new MultipartRequest(req, upPath, fileSize,  "utf-8", new DefaultFileRenamePolicy());    Enumeration<String> fns = multi.getFileNames();  String fileName = "";  String name = "";    if(fns.hasMoreElements()) {  name = fns.nextElement();  fileName = multi.getFilesystemName(name);  }    board.setbNo(multi.getParameter("bNo"));  board.setUserId(multi.getParameter("userId"));  board.setTitle(multi.getParameter("title"));  board.setContent(multi.getParameter("content"));  board.setFileName(fileName);  } catch (Exception e) {  // TODO Auto-generated catch block  e.printStackTrace();  }  return board;  }  } |

### BoardDao.java

|  |
| --- |
| …  public int writeBoard(Board board) {  int result = 0;    String query = "insert into boardTbl values (null, ?, ?, ?, default, ?);";    Connection conn = null;  PreparedStatement pstmt = null;  ResultSet rs = null;    try {  conn = DBConnection.*getConnection*();  pstmt = conn.prepareStatement(query);  pstmt.setString(1, board.getUserId());  pstmt.setString(2, board.getTitle());  pstmt.setString(3, board.getContent());  pstmt.setString(4, board.getFileName());    result = pstmt.executeUpdate();    } catch (ClassNotFoundException | SQLException | NamingException e) {  // TODO Auto-generated catch block  e.printStackTrace();  throw new RuntimeException(e.getMessage());  } finally {  try{  if(pstmt != null){ pstmt.close(); }  if(conn != null){ conn.close(); }  }catch(Exception e){  e.printStackTrace();  throw new RuntimeException(e.getMessage());  }  }  return result;  }  … |

### boardList.jsp – bmsg 처리

|  |
| --- |
| …  <c:choose>  <c:when test=*"*${bmsg == '1'}*"*>  <script>  alert("글이 저장되었습니다.");  </script>  <c:remove var=*"bmsg"* scope=*"session"*/>  </c:when>  <c:when test=*"*${bmsg == '2'}*"*>  <script>  alert("게시글 저장에 실패하였습니다.");  </script>  <c:remove var=*"bmsg"* scope=*"session"*/>  </c:when>  </c:choose>  … |

## 글 내용 보기 처리

### contents.jsp

|  |
| --- |
| <%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*  pageEncoding=*"UTF-8"*%>  <%@ taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"*%>  <!DOCTYPE html>  <html>  <head>  <meta charset=*"UTF-8"*>  <title>글 보기</title>  </head>  <body>  <h3>글 상세 보기</h3>  <form action=*"./updateFrm.bo"* method=*"post"* id=*"data"*>  <table width=*"600"* border=*"3"* bordercolor=*"lightgray"*>  <tr>  <td>작성일</td>  <td>${board.bDate}</td>  </tr>  <tr>  <td>작성자</td>  <td>${board.userId}</td>  </tr>  <tr>  <td>제 목 </td>  <td>${board.title}</td>  </tr>  <tr>  <td>내 용</td>  <td>${board.content}</td>  </tr>  <tr>  <td>첨부파일</td>  <td>  <a href=*"./downfile.bo?fname=*${board.fileName}*"*>${board.fileName}</a>  </td>  </tr>  <tr align=*"center"* valign=*"middle"*>  <td colspan=*"5"*>  <input type=*"button"* value=*"수정"*  onclick=*"updatechk('*${board.userId}*')"*>  <input type=*"button"* value=*"삭제"* onclick=*""*>  <input type=*"button"* value=*"목록"* onclick=*""*>  </td>  </tr>  </table>  </form>  </body>  </html> |

### BoardController.java

|  |
| --- |
| …  else if(command.equals("/content.bo")) {  service = new ContentService();  forward = service.execute(request, response);  }  else if(command.equals("/contentView.bo")) {  forward = new ForwardService();  forward.setRedirect(false);  forward.setPath("/board/content.jsp");  }  else if(command.equals("/downfile.bo")) {  FileProcess.*fileDownload*(request, response);  }  … |

### ContentService.java – 패키지 : com.service.board

|  |
| --- |
| public class ContentService implements Service {  @Override  public ForwardService execute(HttpServletRequest req, HttpServletResponse res)  throws Exception {  ForwardService forward = new ForwardService();    String bNo = req.getParameter("bNo");  String pageNum = req.getParameter("pnum");    BoardDao bDao = BoardDao.getInstance();  Board board = bDao.getBoardContent(bNo);    req.setAttribute("board", board);  req.setAttribute("pnum", pageNum);  forward.setRedirect(false);  forward.setPath("contentView.bo");    return forward;  }  } |

### BoardDao.java

|  |
| --- |
| …  public Board getBoardContent(String bNo) {  Board board = new Board();    String query = "select \* from boardTbl where bNo=?;";    Connection conn = null;  PreparedStatement pstmt = null;  ResultSet rs = null;    try {  conn = DBConnection.*getConnection*();  pstmt = conn.prepareStatement(query);  pstmt.setInt(1, Integer.*parseInt*(bNo));  rs = pstmt.executeQuery();    if(rs.next()) {  board.setbNo(bNo);  board.setTitle(rs.getString("title"));  board.setContent(rs.getString("content"));  board.setUserId(rs.getString("userId"));  board.setbDate(rs.getString("bDate"));  board.setFileName(rs.getString("fileName"));  }  } catch (ClassNotFoundException | SQLException | NamingException e) {  // TODO Auto-generated catch block  e.printStackTrace();  } finally {  // Connection, PreparedStatement를 닫는다.  try{  if(rs != null) { rs.close(); }  if(pstmt != null){ pstmt.close(); }  if(conn != null){ conn.close(); }  }catch(Exception e){  e.printStackTrace();  throw new RuntimeException(e.getMessage());  }  }    return board;  }  … |

### FileProcess.java – 패키지 : com.util

|  |
| --- |
| …  public static int fileDownload(HttpServletRequest req, HttpServletResponse res) {  int result = 0;    String fileName = req.getParameter("fname");  String filePath = req.getServletContext().getRealPath("/board/uploadfiles")  + "/" + fileName;    try {  File file = new File(filePath);  byte b[] = new byte[(int) file.length()];    // page의 ContentType등을 동적으로 바꾸기 위해 초기화시킴  res.reset();  res.setContentType("application/octet-stream");    // 파일 이름 한글 인코딩  String encoding = URLEncoder.*encode*(fileName, "UTF-8");    // 파일 링크를 클릭했을 때 다운로드 저장 화면이 출력되게 처리하는 부분  res.setHeader("Content-Disposition", "attachment;filename="+ encoding);  res.setHeader("Content-Length", String.*valueOf*(file.length()));    if (file.isFile()) { // 파일이 있을경우  FileInputStream fileInputStream = new FileInputStream(file);  ServletOutputStream servletOutputStream = res.getOutputStream();    // 파일을 읽어서 클라이언트쪽으로 저장한다.  int readNum = 0;  while ( (readNum = fileInputStream.read(b)) != -1) {  servletOutputStream.write(b, 0, readNum);  }    servletOutputStream.close();  fileInputStream.close();  }  result = 1;  } catch (Exception e) {  e.printStackTrace();  }  return result;  }  … |

## 글수정 처리

### contents.jsp

|  |
| --- |
| …  <script>  function updatechk(id) {  if(id != '${id}'){  alert('작성자가 아니면 수정할 수 없습니다.');  }  else {  document.getElementById('data').submit();  }  }  </script>  </head>  <body>  <h3>글 상세 보기</h3>  <form action=*"./updateFrm.bo"* method=*"post"* id=*"data"*>  **<input type=*"hidden"* name=*"pnum"* value=*"*${pnum}*"*>**  **<input type=*"hidden"* name=*"bNo"* value=*"*${board.bNo}*"*>**  **<input type=*"hidden"* name=*"userId"* value=*"*${board.userId}*"*>**  **<input type=*"hidden"* name=*"title"* value=*"*${board.title}*"*>**  **<input type=*"hidden"* name=*"content"* value=*"*${board.content}*"*>**  **<input type=*"hidden"* name=*"fileName"* value=*"*${board.fileName}*"*>**  <table width=*"600"* border=*"3"* bordercolor=*"lightgray"*>  …  <tr align=*"center"* valign=*"middle"*>  <td colspan=*"5"*>  <input type=*"button"* value=*"수정"*  onclick=*"updatechk('*${board.userId}*')"*>  <input type=*"button"* value=*"삭제"* onclick=*""*>  <input type=*"button"* value=*"목록"*  onclick=*"location.href='./list.bo?pnum=*${pnum}*'"*>  </td>  </tr>  … |

### updateFrm.jsp

|  |
| --- |
| <%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*  pageEncoding=*"UTF-8"*%>  <!DOCTYPE html>  <html>  <head>  <meta charset=*"UTF-8"*>  <title>글수정</title>  </head>  <body>  <h3>게시글 수정</h3>  <form action=*"./update.bo?pnum=*${param.pnum}*"* method=*"post"* enctype=*"multipart/form-data"*>  <input type=*"hidden"* name=*"bNo"* value=*"*${param.bNo}*"*>  <table>  <tr>  <td width=*"100"*>아이디</td>  <td><input type=*"text"* name=*"userId"* value=*"*${param.userId}*"*  readonly=*"readonly"*></td>  </tr>  <tr>  <td>제목</td>  <td><input type=*"text"* name=*"title"* value=*"*${param.title}*"* size=*"60"*  required=*"required"*>  </td>  </tr>  <tr>  <td>내용</td>  <td><textarea name=*"content"* cols=*"62"* rows=*"20"*>${param.content}  </textarea></td>  </tr>  <tr>  <td>기존 파일</td>  <td>${empty param.fileName ? '없음' : param.fileName}</td>  </tr>  <tr>  <td>파일 첨부</td>  <td><input type=*"file"* name=*"file"*></td>  </tr>  </table>  <br>  <input type=*"submit"* value=*"수정"* />  <input type=*"button"* value=*"취소"* onclick="history.back()">  </form>  </body>  </html> |

### BoardController.java

|  |
| --- |
| …  else if(command.equals("/updateFrm.bo")) {  request.setCharacterEncoding("UTF-8");  forward = new ForwardService();  forward.setRedirect(false);  forward.setPath("/board/updateFrm.jsp");  }  else if(command.equals("/update.bo")) {  service = new UpdateService();  forward = service.execute(request, response);  }  … |

### UpdateService.java – 패키지 : com.service.board

|  |
| --- |
| public class UpdateService implements Service {  @Override  public ForwardService execute(HttpServletRequest req, HttpServletResponse res)  throws Exception {  ForwardService forward = new ForwardService();  HttpSession session = req.getSession();  req.setCharacterEncoding("UTF-8");  String pageNum = req.getParameter("pnum");    Board board = FileProcess.fileUpload(req);  BoardDao dDao = BoardDao.getInstance();  int result = dDao.updateBoard(board);    if(result > 0) {  session.setAttribute("bmsg", "3");  }  else {  session.setAttribute("bmsg", "4");  }    forward.setRedirect(true);  forward.setPath("./list.bo?pnum=" + pageNum);    return forward;  }  } |

### BoardDao.java

|  |
| --- |
| …  public int updateBoard(Board board) throws SQLException {  int result = 0;  String query = "update boardTbl set title=?, content=?, ";    if(board.getFileName() != null) {  query += "fileName=?, ";  }  query += "bDate=now() where bNo=?;";    Connection conn = null;  PreparedStatement pstmt = null;  ResultSet rs = null;    try {  conn = DBConnection.*getConnection*();  conn.setAutoCommit(false);    pstmt = conn.prepareStatement(query);  pstmt.setString(1, board.getTitle());  pstmt.setString(2, board.getContent());  if(board.getFileName() != null) {  pstmt.setString(3, board.getFileName());  pstmt.setString(4, board.getbNo());  }  else {  pstmt.setString(3, board.getbNo());  }    pstmt.executeUpdate();    conn.commit();  result = 1;  } catch (ClassNotFoundException | SQLException | NamingException e) {  // TODO Auto-generated catch block  e.printStackTrace();  conn.rollback();  result = 0;  } finally {  // Connection, PreparedStatement를 닫는다.  try{  if(pstmt != null){ pstmt.close(); }  if(conn != null){ conn.close(); }  }catch(Exception e){  e.printStackTrace();  }  }  return result;  }  … |

### boardList.jsp – bmsg 처리

|  |
| --- |
| …  <c:choose>  <c:when test=*"*${bmsg == '3'}*"*>  <script>  alert("글을 수정하였습니다.");  </script>  <c:remove var=*"bmsg"* scope=*"session"*/>  </c:when>  <c:when test=*"*${bmsg == '4'}*"*>  <script>  alert("게시글 수정에 실패하였습니다.");  </script>  <c:remove var=*"bmsg"* scope=*"session"*/>  </c:when>  </c:choose>  … |

## 글삭제 처리

### content.jsp – script 추가, 삭제 버튼 수정

|  |
| --- |
| function deletechk(id, bnum){  if(id != '${id}'){  alert('작성자가 아니면 삭제할 수 없습니다.');  }  else {  var res = confirm('글을 삭제하시겠습니까?');  if(res){  location.href='./delete.bo?bNo=' + bnum;  }  }  }  …  <input type=*"button"* value=*"삭제"* onclick=*"deletechk('*${board.userId}*',*${board.bNo}*)"*>  … |

### BoardController.java

|  |
| --- |
| …  else if(command.equals("/delete.bo")) {  service = new DeleteService();  forward = service.execute(request, response);  }  … |

### DeleteService.java – 패키지 : com.service.board

|  |
| --- |
| public class DeleteService implements Service {  @Override  public ForwardService execute(HttpServletRequest req, HttpServletResponse res)  throws Exception {  ForwardService forward = new ForwardService();  HttpSession session = req.getSession();  String filePath = req.getServletContext().getRealPath("/board/uploadfiles");  String bNo = req.getParameter("bNo");  int bNum = 0;    if(bNo != null) {  bNum = Integer.parseInt(bNo);  }    BoardDao bDao = BoardDao.getInstance();  int result = bDao.deleteBoard(bNum, filePath);    if(result == 1) {  session.setAttribute("bmsg", "5");  }  else {  session.setAttribute("bmsg", "6");  }    forward.setRedirect(true);  forward.setPath("./list.bo?pnum=1");    return forward;  }  } |

### BoardDao.java

|  |
| --- |
| …  public int deleteBoard(int bNum, String filePath) throws SQLException {  int result = 0;    String query = "delete from boardTbl where bNo=?;";    Connection conn = null;  PreparedStatement pstmt = null;    try {  conn = DBConnection.*getConnection*();  conn.setAutoCommit(false);    pstmt = conn.prepareStatement(query);  pstmt.setInt(1, bNum);  pstmt.executeUpdate();    String fileName = getFileName(bNum);    if(fileName != null && !fileName.equals("")) {  FileProcess.*deleteFile*(filePath + "/" + fileName);  }    conn.commit();  result = 1;  } catch (Exception e) {  e.printStackTrace();  conn.rollback();  result = 0;  } finally {  try{  if(pstmt != null){ pstmt.close(); }  if(conn != null){ conn.close(); }  }catch(Exception e){  e.printStackTrace();  }  }  return result;  }  private String getFileName(int bNum) {  String fileName = "";    String query = "select fileName from boardTbl where bNo=?;";    Connection conn = null;  PreparedStatement pstmt = null;  ResultSet rs = null;    try {  conn = DBConnection.*getConnection*();  pstmt = conn.prepareStatement(query);  pstmt.setInt(1, bNum);  rs = pstmt.executeQuery();  if(rs.next()) {  fileName = rs.getString("fileName");  }  } catch (ClassNotFoundException | SQLException | NamingException e) {  // TODO Auto-generated catch block  e.printStackTrace();  } finally {  // Connection, PreparedStatement를 닫는다.  try{  if(rs != null) { rs.close(); }  if(pstmt != null){ pstmt.close(); }  if(conn != null){ conn.close(); }  }catch(Exception e){  e.printStackTrace();  }  }  return fileName;  }  … |

### FileProcess.java – 패키지 : com.util

|  |
| --- |
| …  public static void deleteFile(String fileName){  System.*out*.println("delete file: " + fileName);  File file = new File(fileName);    if(file.exists()) {  if(!file.delete()) {  throw new RuntimeException();  }  }  else {  throw new RuntimeException();  }  }  … |

### boardList.jsp – bmsg 처리

|  |
| --- |
| …  <c:choose>  <c:when test=*"*${bmsg == '5'}*"*>  <script>  alert("글을 수정하였습니다.");  </script>  <c:remove var=*"bmsg"* scope=*"session"*/>  </c:when>  <c:when test=*"*${bmsg == '6'}*"*>  <script>  alert("게시글 수정에 실패하였습니다.");  </script>  <c:remove var=*"bmsg"* scope=*"session"*/>  </c:when>  </c:choose>  … |

## 검색 기능 추가

### boardList.jsp – 환영 테이블과 글목록 테이블 사이에 추가

|  |
| --- |
| …  <form action=*"./list.bo?pnum=1"* method=*"post"*>  <select name=*"opt"*>  <option value=*"0"*>제목</option>  <option value=*"1"*>내용</option>  <option value=*"2"*>제목+내용</option>  <option value=*"3"*>작성자</option>  </select>  <input type=*"text"* size=*"20"* name=*"condition"*/>  <input type=*"submit"* value=*"검색"*>  </form>  … |

### ListService.java – 페이지 번호 추출과 페이지 처리 사이(페이지번호, 검색관련 값을 HashMap으로 처리)

|  |
| --- |
| … //condition 값 한글 처리를 위해 위쪽에 req.setCharacterEncoding(“UTF-8”); 추가  String opt = req.getParameter("opt");  String condition = req.getParameter("condition");  HashMap<String, Object> listOpt = new HashMap<>();  listOpt.put("opt", opt);  listOpt.put("condition", condition);  listOpt.put("pageNum", pageNum);    BoardDao bDao = BoardDao.*getInstance*();  int listCnt = bDao.getBoardListCnt(**listOpt**);  ArrayList<Board> bList = bDao.getBoardList(**listOpt**);  … |

### BoardDao.java – getBoardListCnt : HashMap으로 처리하고 검색 관련 기능 추가

|  |
| --- |
| public int getBoardListCnt(HashMap<String, Object> listOpt) {  int result = 0;    String opt = (String)listOpt.get("opt");  String condition = "%" + (String)listOpt.get("condition") + "%";    String query = "select count(\*) as cnt from boardTbl";    Connection conn = null;  PreparedStatement pstmt = null;  ResultSet rs = null;    try {  conn = DBConnection.*getConnection*();  if(opt == null) {  query += ";";  pstmt = conn.prepareStatement(query);  }else if(opt.equals("0")) { //제목 검색  query += " where title like ?;";  pstmt = conn.prepareStatement(query);  pstmt.setString(1, condition);  }else if(opt.equals("1")) { //내용 검색  query += " where content like ?;";  pstmt = conn.prepareStatement(query);  pstmt.setString(1, condition);  }else if(opt.equals("2")) { //제목+내용 검색  query += " where title like ? or content like ?;";  pstmt = conn.prepareStatement(query);  pstmt.setString(1, condition);  pstmt.setString(2, condition);  }else if(opt.equals("3")) { //작성자  query += " where userId like ?;";  pstmt = conn.prepareStatement(query);  pstmt.setString(1, condition);  }  rs = pstmt.executeQuery();    if(rs.next()) { result = rs.getInt("cnt"); }  } catch (ClassNotFoundException | SQLException | NamingException e) {  e.printStackTrace();  } finally {  try{  if(rs != null) { rs.close(); }  if(pstmt != null){ pstmt.close(); }  if(conn != null){ conn.close(); }  }catch(Exception e){  e.printStackTrace();  }  }  return result;  } |

### BoardDao.java – getBoardList : HashMap으로 처리하고 검색 관련 기능 추가

|  |
| --- |
| public ArrayList<Board> getBoardList(HashMap<String, Object> listOpt){  ArrayList<Board> bList = new ArrayList<>();    String opt = (String)listOpt.get("opt");  String condition = "%" + (String)listOpt.get("condition") + "%";  int pageNum = (int)listOpt.get("pageNum");    int limitNum = (pageNum - 1) \* 10;//페이지 번호로 limit 값 계산    String query = "select \* from boardTbl";  String query2 = " order by bNo DESC limit ?, 10;";    Connection conn = null;  PreparedStatement pstmt = null;//파라미터 있는 SQL 쿼리문 실행용  ResultSet rs = null;  try {  conn = DBConnection.*getConnection*();    if(opt == null) {  pstmt = conn.prepareStatement(query + query2);  pstmt.setInt(1, limitNum);  }else if(opt.equals("0")) {  query += " where title like ?";  pstmt = conn.prepareStatement(query + query2);  pstmt.setString(1, condition);  pstmt.setInt(2, limitNum);  }else if(opt.equals("1")) {  query += " where content like ?";  pstmt = conn.prepareStatement(query + query2);  pstmt.setString(1, condition);  pstmt.setInt(2, limitNum);  }else if(opt.equals("2")) {  query += " where title like ? or content like ?";  pstmt = conn.prepareStatement(query + query2);  pstmt.setString(1, condition);  pstmt.setString(2, condition);  pstmt.setInt(3, limitNum);  }else if(opt.equals("3")) {  query += " where userId like ?";  pstmt = conn.prepareStatement(query + query2);  pstmt.setString(1, condition);  pstmt.setInt(2, limitNum);  }  rs = pstmt.executeQuery();    while(rs.next()) {  Board board = new Board();  board.setbNo(String.*valueOf*(rs.getInt("bNo")));  board.setUserId(rs.getString("userId"));  board.setTitle(rs.getString("title"));  board.setContent(rs.getString("content"));  board.setbDate(rs.getString("bDate"));  bList.add(board);  }  } catch (ClassNotFoundException | SQLException | NamingException e) {  e.printStackTrace();  } finally {  try{  if(rs != null) { rs.close(); }  if(pstmt != null){ pstmt.close(); }  if(conn != null){ conn.close(); }  }catch(Exception e){  e.printStackTrace();  }  }  return bList;  } |