

http://localhost:8080/ProjectBoard\_Completed/list

http://localhost:8080/ProjectBoard_Completed/list				
번호	제목	작성자	작성일시	조회수
25	글23	장영실	2021-02-24 14:19:40	0
24	글22	래리페이지	2021-02-24 14:19:40	0
23	글21	스티브잡스	2021-02-24 14:19:40	0
22	글20	빌 게이츠	2021-02-24 14:19:40	0
21	글19	정약용	2021-02-24 14:19:40	0
1 2 3 다음				
글쓰기				

http://localhost:8080/ProjectBoard_Completed/list?page=4				
번호	제목	작성자	작성일시	조회수
10	글8	이순신	2021-02-24 14:14:12	0
7	글7	홍길동	2021-02-17 20:09:24	1
6	글6	김수로	2021-02-17 20:09:24	0
5	글5	장길산	2021-02-17 20:09:24	2
4	글4	김수로	2021-02-24 13:32:34	1
이전 4 5				
글쓰기				

http://localhost:8080/ProjectBoard_Completed/write	
제목	<input type="text"/>
작성자	<input type="text"/>
내용	<div><div></div><div>↑</div><div>↓</div></div>
저장	취소

➤ 내용 작성 후 저장 시 list로 페이지가 이동함

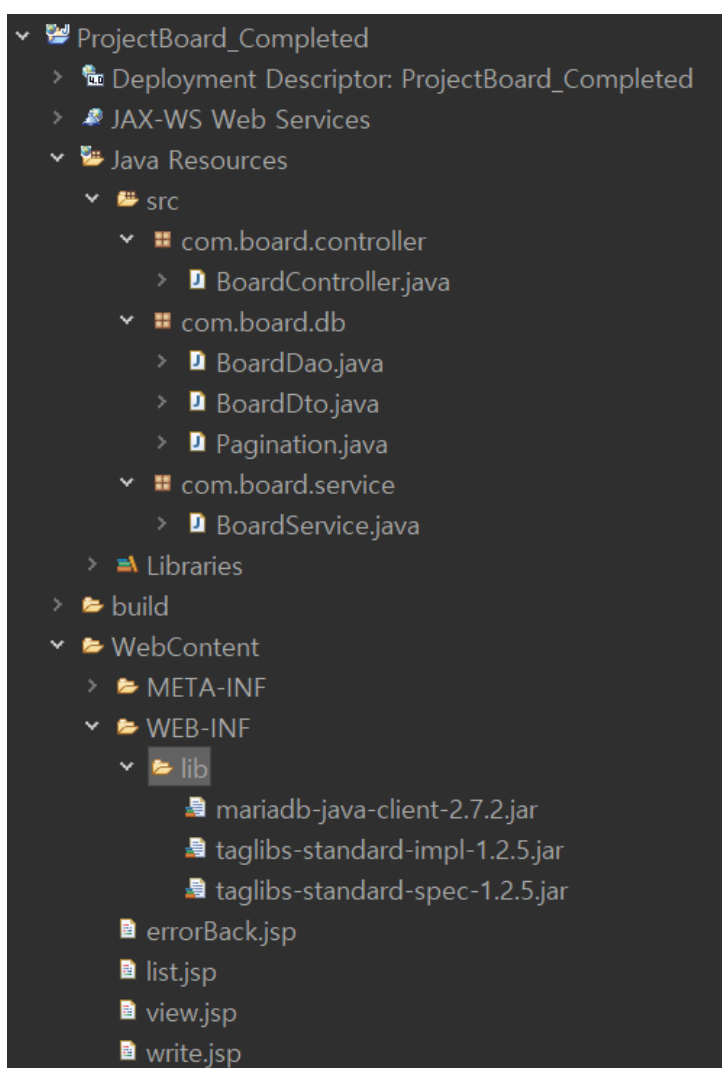
← → ↻ ↺ http://localhost:8080/ProjectBoard\_Completed/view?num=10

제목	글8
작성자	이순신
작성일시	2021-02-24 14:14:12
조회수	0
내용	글의 내용8

목록보기

수정

삭제



**<!-- List.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">
```

```
    <style>
```

```
        table { width:680px; text-align:center; }
```

```
        th { background-color:cyan; }
```

```
        .num { width:80px; }
```

```
        .title { width:230px; }
```

```
        .writer { width:100px; }
```

```
        .regtime { width:180px; }
```

```
        a:link { text-decoration:none; color:blue }
```

```
        a:visited { text-decoration:none; color:gray }
```

```
        a:hover { text-decoration:none; color:red }
```

```
    </style>
```

```
</head>
```

```
<body>
```

```
    <table>
```

```
        <tr>
```

```
            <th class="num">번호</th>
```

```
            <th class="title">제목</th>
```

```
            <th class="writer">작성자</th>
```

```
            <th class="regtime">작성일시</th>
```

```
            <th>조회수</th>
```

```
        </tr>
```

```
        <c:forEach var="msg" items="${msgList}">
```

```
            <tr>
```

```
                <td>${msg.num}</td>
```

```
                <td style="text-align:left;">
```

```
                    <a href="view?num=${msg.num}">${msg.title}</a>
```

```
                </td>
```

```
                <td>${msg.writer}</td>
```

```
                <td>${msg.regTime}</td>
```

```
                <td>${msg.hits}</td>
```

```
            </tr>
```

```
        </c:forEach>
```

```
    </table>
```

```
    <br>
```

```
<div style="width:680px; text-align:center;">
  <c:forEach var="pgn" items="${pgnList}">
    <a class="pgn" href="list?page=${pgn.pageNum}">
      <c:choose>
        <c:when test="${pgn.curPage}">
          <u>${pgn.display}</u>
        </c:when>
        <c:otherwise>
          ${pgn.display}
        </c:otherwise>
      </c:choose>
    </a>&nbsp;
  </c:forEach>
</div>

<input type="button" value="글쓰기" onclick="location.href='write'">

</body>
</html>
```

```
<!-- view.jsp -->
```

```
<%@ page language="java" contentType="text/html; charset=utf-8" pageEncoding="utf-8" %>
<%@ page import = "com.board.db.*" %>
```

```
<!doctype html>
```

```
<html>
```

```
<head>
```

```
    <meta charset="utf-8">
```

```
    <style>
```

```
        table { width:680px; text-align:center; }
```

```
        th { width:100px; background-color:cyan; }
```

```
        td { text-align:left; border:1px solid gray; }
```

```
    </style>
```

```
</head>
```

```
<body>
```

```
    <table>
```

```
        <tr>
```

```
            <th>제목</th>
```

```
            <td>${msg.title}</td>
```

```
        </tr>
```

```
        <tr>
```

```
            <th>작성자</th>
```

```
            <td>${msg.writer}</td>
```

```
        </tr>
```

```
        <tr>
```

```
            <th>작성일시</th>
```

```
            <td>${msg.regTime}</td>
```

```
        </tr>
```

```
        <tr>
```

```
            <th>조회수</th>
```

```
            <td>${msg.hits}</td>
```

```
        </tr>
```

```
        <tr>
```

```
            <th>내용</th>
```

```
            <td>${msg.content}</td>
```

```
        </tr>
```

```
    </table><br><br>
```

```
    <input type="button" value="목록보기" onclick="location.href='list'">
```

```
    <input type="button" value="수정" onclick="location.href='write?num=${msg.num}'">
```

```
    <input type="button" value="삭제" onclick="location.href='delete?num=${msg.num}'">
```

```
</body>
```

```
</html>
```

```
<!-- write.jsp -->
```

```
<%@ page language="java" contentType="text/html; charset=utf-8" pageEncoding="utf-8" %>
<%@ page import = " com.board.db.*" %>
```

```
<!doctype html>
```

```
<html>
```

```
<head>
```

```
    <meta charset="utf-8">
```

```
    <style>
```

```
        table { width:680px; text-align:center; }
```

```
        th { width:100px; background-color:cyan; }
```

```
        input[type=text], textarea { width:100%; }
```

```
    </style>
```

```
</head>
```

```
<body>
```

```
    <form action="${action}" method="post">
```

```
        <table>
```

```
            <tr>
```

```
                <th>제목</th>
```

```
                <td>
```

```
                    <input type="text" name="title" maxlength="80" value="${msg.title}"></td>
```

```
            </tr>
```

```
            <tr>
```

```
                <th>작성자</th>
```

```
                <td>
```

```
                    <input type="text" name="writer" maxlength="20" value="${msg.writer}">
```

```
                </td>
```

```
            </tr>
```

```
            <tr>
```

```
                <th>내용</th>
```

```
                <td>
```

```
                    <textarea name="content" rows="10">${msg.content}</textarea>
```

```
                </td>
```

```
            </tr>
```

```
        </table>
```

```
        <br>
```

```
        <input type="submit" value="저장">
```

```
        <input type="button" value="취소" onclick="history.back()">
```

```
    </form>
```

```
</body>
```

```
</html>
```

```
<!-- errorBack.jsp -->
```

```
<%@ page language="java" contentType="text/html; charset=utf-8"  
    pageEncoding="utf-8" %>
```

```
<!doctype html>
```

```
<html>
```

```
<head>
```

```
    <meta charset="utf-8">
```

```
</head>
```

```
<body>
```

```
    <script>
```

```
        alert('${errorMessage}');
```

```
        history.back();
```

```
    </script>
```

```
</body>
```

```
</html>
```

```
<!-- BoardDto.java -->
```

```
package com.board.db;
```

```
public class BoardDto {
```

```
    private int num = 0;
    private String title = "";
    private String writer = "";
    private String content = "";
    private String regTime = "";
    private int hits = 0;
```

```
    public int getNum() { return num; }
    public void setNum(int num) {
        this.num = num;
    }
```

```
    public String getTitle() { return title; }
    public void setTitle(String title) {
        this.title = title;
    }
```

```
    public String getWriter() { return writer; }
    public void setWriter(String writer) {
        this.writer = writer;
    }
```

```
    public String getContent() { return content; }
    public void setContent(String content) {
        this.content = content;
    }
```

```
    public String getRegTime() { return regTime; }
    public void setRegTime(String regTime) {
        this.regTime = regTime;
    }
```

```
    public int getHits() { return hits; }
    public void setHits(int hits) {
        this.hits = hits;
    }
```

```
}
```



```
<!-- BoardDao.java -->
```

```
package com.board.db;
```

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

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

```
import java.time.*;
```

```
public class BoardDao {
```

```
    private Connection myConnection() throws Exception {
        Class.forName("org.mariadb.jdbc.Driver");
        Connection con = DriverManager.getConnection(
            "jdbc:mariadb://localhost:3306/infodb", "jspGuest", "1234"
        );
        return con;
    }
```

```
    private String getCurTime() {
        return LocalDate.now() + " " + LocalTime.now().toString().substring(0, 8);
    }
```

```
    public int getNumRecords() {
        int numRecords = 0;

        try (
            Connection con = myConnection();
            Statement stmt = con.createStatement();
            ResultSet rs = stmt.executeQuery("select count(*) from boardtbl");
        ) {
            if (rs.next()) {
                numRecords = rs.getInt(1);
            }
        }
        catch (Exception e){
            e.printStackTrace();
        }
        return numRecords;
    }
```

```
    public ArrayList<BoardDto> selectList(int start, int listSize) {
        ArrayList<BoardDto> dtoList = new ArrayList<BoardDto>();
        try (
            Connection con = myConnection();
            Statement stmt = con.createStatement();
            ResultSet rs = stmt.executeQuery(String.format(
                "select * from boardtbl order by num desc limit %d, %d",
                start, listSize
            ));
        ) {
            while (rs.next()) {
                BoardDto dto = new BoardDto();
                dto.setNum(rs.getInt("num"));
                dto.setTitle(rs.getString("title"));
                dto.setContent(rs.getString("content"));
                dtoList.add(dto);
            }
        }
        return dtoList;
    }
```

```

        ));
    } {
        while(rs.next()) {
            BoardDto dto = new BoardDto();
            dto.setNum(rs.getInt("num"));
            dto.setTitle(rs.getString("title"));
            dto.setWriter(rs.getString("writer"));
            dto.setContent(rs.getString("content"));
            dto.setRegTime(rs.getString("regTime"));
            dto.setHits(rs.getInt("hits"));
            dtoList.add(dto);
        }
    }
    catch (Exception e) {
        e.printStackTrace();
    }
    return dtoList;
}

```

```

public BoardDto selectOne(int num, boolean increaseHits) {
    BoardDto dto = new BoardDto();

    try (
        Connection con = myConnection();
        Statement stmt = con.createStatement();
        ResultSet rs = stmt.executeQuery(
            "select * from boardtbl where num = " + num
        );
    ) {
        if(rs.next()) {
            dto.setNum(num);
            dto.setTitle(rs.getString("title"));
            dto.setWriter(rs.getString("writer"));
            dto.setContent(rs.getString("content"));
            dto.setRegTime(rs.getString("regTime"));
            dto.setHits(rs.getInt("hits"));

            if(increaseHits) {
                stmt.executeUpdate(
                    "update boardtbl set hits = hits+1 where num = " + num
                );
            }
        }
    }
    catch (Exception e) {
        e.printStackTrace();
    }

    return dto;
}

```

```

public void insertOne(BoardDto dto) {
    try(
        Connection con = myConnection();
        Statement stmt = con.createStatement();
    ) {
        stmt.executeUpdate(String.format(
            "insert into boardtbl values (null, '%s', '%s', '%s', '%s', 0)",
            dto.getWriter(), dto.getTitle(), dto.getContent(), getCurTime()
        ));
    }
    catch (Exception e) {
        e.printStackTrace();
    }
}

```

```

public void updateOne(BoardDto dto) {
    try (
        Connection con = myConnection();
        Statement stmt = con.createStatement();
    ) {
        stmt.executeUpdate(String.format(
            "update boardtbl set title = '%s', writer='%s', content='%s', regTime='%s'"
            + "where num = '%d'", dto.getTitle(), dto.getWriter(), dto.getContent(),
            getCurTime(), dto.getNum()
        ));
    }
    catch (Exception e) {
        e.printStackTrace();
    }
}

```

```

public void deleteOne(int num) {
    try (
        Connection con = myConnection();
        Statement stmt = con.createStatement();
    ) {
        stmt.executeUpdate("delete from boardtbl where num=" + num);
    }
    catch (Exception e) {
        e.printStackTrace();
    }
}

```

```

}

```

```
<!-- Pagination.java -->
```

```
package com.board.db;
```

```
public class Pagination {
```

```
    private String display;  
    private int pageNum;  
    private boolean curPage;
```

```
    public Pagination(String display, int pageNum, boolean curPage) {  
        this.display = display;  
        this.pageNum = pageNum;  
        this.curPage = curPage;  
    }
```

```
    public String getDisplay() {  
        return display;  
    }
```

```
    public void setDisplay(String display) {  
        this.display = display;  
    }
```

```
    public int getPageNum() {  
        return pageNum;  
    }
```

```
    public void setPageNum(int pageNum) {  
        this.pageNum = pageNum;  
    }
```

```
    public boolean isCurPage() {  
        return curPage;  
    }
```

```
    public void setCurPage(boolean curPage) {  
        this.curPage = curPage;  
    }
```

```
}
```

*<!-- BoardController.java -->*

```
package com.board.controller;

import java.io.IOException;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.board.db.*;
import com.board.service.*;

@WebServlet("/")
public class BoardController extends HttpServlet {

    private static final long serialVersionUID = 1L;

    public BoardController() { super(); }

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        String view = null;

        String uri = request.getRequestURI();
        String conPath = request.getContextPath();
        String com = uri.substring(conPath.length());

        if (com.equals("/list") || com.equals("/")) {
            String tmp = request.getParameter("page");
            int pageNum = (tmp!=null && tmp.length()>0) ? Integer.parseInt(tmp) : 1;
            request.setAttribute("msgList", new BoardService().getMsgList(pageNum));
            request.setAttribute("pgnList", new BoardService().getPagination(pageNum));
            view = "list.jsp";
        }

        else if (com.equals("/view")) {
            int num = Integer.parseInt(request.getParameter("num"));
            request.setAttribute("msg", new BoardService().getMsg(num));
            view = "view.jsp";
        }

        else if (com.equals("/write")) {
            BoardDto dto = new BoardDto();
            String action = "insert";

            String tmp = request.getParameter("num");
            int num = (tmp != null && tmp.length() > 0) ? Integer.parseInt(tmp) : 0;
```

```

    if (num > 0) {
        dto = new BoardService().getMsgForWrite(num);
        action = "update?num=" + num;
    }

    request.setAttribute("msg", dto);
    request.setAttribute("action", action);
    view = "write.jsp";
}

else if (com.equals("/insert")) {
    request.setCharacterEncoding("utf-8");
    String writer = request.getParameter("writer");
    String title = request.getParameter("title");
    String content = request.getParameter("content");

    try {
        new BoardService().writeMsg(writer, title, content);
        view = "redirect:list";
    }
    catch (Exception e) {
        request.setAttribute("errorMessage", "모든 항목을 입력하세요.");
        view = "errorBack.jsp";
    }
}

else if (com.equals("/update")) {
    request.setCharacterEncoding("utf-8");
    int num = Integer.parseInt(request.getParameter("num"));

    String writer = request.getParameter("writer");
    String title = request.getParameter("title");
    String content = request.getParameter("content");

    try {
        new BoardService().updateMsg(num, writer, title, content);
        view = "redirect:list";
    }
    catch (Exception e) {
        request.setAttribute("errorMessage", "모든 항목을 입력하세요.");
        view = "errorBack.jsp";
    }
}

else if (com.equals("/delete")) {
    int num = Integer.parseInt(request.getParameter("num"));
    new BoardService().deleteMsg(num);
    view="redirect:list";
}

```

```
        if (view.startsWith("redirect:")) {
            response.sendRedirect(view.substring(9));
        }
        else {
            RequestDispatcher dispatcher = request.getRequestDispatcher(view);
            dispatcher.forward(request, response);
        }
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        doGet(request, response);
    }
}
```

```
<!-- BoardService.java -->
```

```
package com.board.service;
```

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

```
import com.board.db.*;
```

```
public class BoardService {
```

```
    private int listSize = 5;
```

```
    private static final int paginationSize = 3;
```

```
    public ArrayList<BoardDto> getMsgList(int pageNum) {  
        return new BoardDao().selectList((pageNum-1)*listSize, listSize);  
    }
```

```
    public ArrayList<Pagination> getPagination(int pageNum) {
```

```
        ArrayList<Pagination> pgnList = new ArrayList<Pagination>();
```

```
        int numRecords = new BoardDao().getNumRecords();
```

```
        int numPages = (int)Math.ceil((double)numRecords / listSize);
```

```
        int firstLink = ((pageNum - 1) / paginationSize) * paginationSize + 1;
```

```
        int lastLink = firstLink + paginationSize - 1;
```

```
        if (lastLink > numPages) { lastLink = numPages; }
```

```
        if (firstLink > 1) {
```

```
            pgnList.add(new Pagination("이전", pageNum - paginationSize, false));
```

```
        }
```

```
        for (int i = firstLink; i<=lastLink; i++) {
```

```
            pgnList.add(new Pagination("" + i, i, i == pageNum));
```

```
        }
```

```
        if (lastLink < numPages) {
```

```
            int tmpPageNum = pageNum + paginationSize;
```

```
            if (tmpPageNum > numPages) {
```

```
                tmpPageNum = numPages;
```

```
            }
```

```
            pgnList.add(new Pagination("다음", tmpPageNum, false));
```

```
        }
```

```
        return pgnList;
```

```
    }
```



```

public BoardDto getMsg(int num) {
    BoardDto dto = new BoardDao().selectOne(num, true);

    dto.setTitle(dto.getTitle().replace(" ", "&nbsp;"));
    dto.setContent(
        dto.getContent().replace(" ", "&nbsp;").replace("\n", "<br>"));

    return dto;
}

public BoardDto getMsgForWrite(int num) {
    return new BoardDao().selectOne(num, false);
}

public boolean checkNull(String writer, String title, String content) {
    if (writer == null || writer.length() == 0 ||
        title == null || title.length() == 0 ||
        content == null || content.length() == 0) {
        return true;
    }
    return false;
}

public void writeMsg(String writer, String title, String content)
    throws Exception {
    if (checkNull(writer, title, content)) {
        throw new Exception("모든 항목을 입력하세요.");
    }
    BoardDto dto = new BoardDto();
    dto.setWriter(writer);
    dto.setTitle(title);
    dto.setContent(content);

    new BoardDao().insertOne(dto);
}

public void updateMsg(int num, String writer, String title, String content)
    throws Exception {
    if (checkNull(writer, title, content)) {
        throw new Exception ("모든 항목을 입력하세요.");
    }
    BoardDto dto = new BoardDto();
    dto.setNum(num);
    dto.setWriter(writer);
    dto.setTitle(title);
    dto.setContent(content);

    new BoardDao().updateOne(dto);
}

```

```
public void deleteMsg(int num) {  
    new BoardDao().deleteOne(num);  
}  
  
}
```