

Spring 게시판 프로젝트



목 차



JSP

실제 페이지 부분



Controller

요청이 있을 시 페이지
연결부분



DATA

ORACLE 데이터와
연결되는 부분



Service 객체부분

객체 생성 부분

실제 페이지 부분



JSP

CONTENTS Board.jsp - 게시판 구현부분



```
1 <%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" language="java" %>
2 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
3 <!DOCTYPE html>
4 <html>
5 <head>
6 <title>Board</title>
7 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-1BmE4kWbQ78iYhFLdvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
8   crossorigin="anonymous">
9 </head>
10
11 <body style="background-color: #17a2b7;">
12 <header>
13 <div>
14   <%@ include file="./Navbar.jsp"%>
15 </div>
16 </header>
17 <!-- -----헤더와 세션 구분선----->
18 <section style = "height:700px; width:1200px;">
19 <div class="container" style="position:relative; left:140px; top: 40px;">
20 <table class="table" style="width:1200px;">
21 <tr>
22 <th scope="col">번호</th>
23 <th scope="col">이름</th>
24 <th scope="col">제목</th>
25 <th scope="col">날짜</th>
26 <th scope="col">조회수</th>
27 </tr>
28 <c:forEach items="${board}" var="dto">
29 <tr>
30 <th scope="col">${dto.bId}</th>
31 <th scope="col">${dto.bName}</th>
32 <th scope="col">
33 <c:forEach begin="1" end="${dto.bIndent}">-</c:forEach>
34 <a href="board_manage?bId=${dto.bId}">${dto.bTitle}</a></th>
35 <th scope="col">${dto.bDate}</th>
36 <th scope="col">${dto.bHit}</th>
37 </tr>
38 </c:forEach>
39 </table>
40 <div style="position:relative; left:30px; top: 40px;" class="text-end"> <a class="btn btn-primary" href="board_write" role="button">글작성</a></div>
41 </div>
42 </section>
43 <!-- -----세션과 푸터 구분선 ----->
44 <footer>
45 <hr>
46 <div class="container" style="color:white;">
47 <div class="text-center">
48   안녕하세요.부족하지만 최선을 다하는 프로그래머 이승훈 입니다.
49 </div>
50 <div class="text-end" style="padding-top:30px;">
51   주소 : 대구광역시 남구 대명동<br>
52   전화번호 : 010-3926-1685
53 </div>
54 </div>
55 </footer>
56
57 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYs0g+OMhuP+ILRH9sENB00LRn5q+8nbTov4+1p"
58   crossorigin="anonymous"></script>
59 </body>
60 </html>
```

Windows 상품 인증

[설정]으로 이동하여 Windows를 정품 인증합니다.

CONTENTS Board_Write.jsp - 게시판 구현부분



```
<section style = "height:700px; width:1200px;">
<div class="container" style="position:relative; top: 40px; left:140px;">
  <table class="table" style="width:1200px;" >
    <form action="write" method="post">
      <tr class="row g-2">
        <th class="col-md-2"> 이름 </th>
        <th class="col-md-10">
          <input type="text" name="bName" class="form-control" placeholder="이름" aria-label="이름" aria-describedby="basic-addon1">
        </th>
      </tr>
      <tr class="row g-2">
        <th class="col-md-2"> 제목 </th>
        <th class="col-md-10">
          <input type="text" name="bTitle" class="form-control" placeholder="제목" aria-label="제목" aria-describedby="basic-addon1">
        </th>
      </tr>
      <tr class="row g-2">
        <th class="col-md-2"> 내용 </th>
        <th class="col-md-10">
          <div class="form-floating">
            <textarea name="bContent" class="form-control" style="height: 400px"></textarea>
          </div>
        </th>
      </tr>
      <tr >
        <th class="col-md-12 text-end">
          <input type="submit" value="입력" class="btn btn-primary">
          <a href="board.do" class="btn btn-primary">목록보기</a>
        </th>
      </tr>
    </form>
  </table>
</div>
</section>
```

CONTENTS Board_Manage.jsp - 게시판 구현부분

```
<section style = "height:750px; width:1200px;">
<div class="container" style="position:relative; top: 40px; left:140px;">
    <table class="table" style="width:1200px;" >
        <form action="modify" method="post">
            <input type="hidden" name="bId" value="{board_view.bId}">
            <tr class="row_g-2">
                <th class="col-md-2"> 번호 </th>
                <th class="col-md-10"> {board_view.bId} </th>
            </tr>
            <tr class="row_g-2">
                <th class="col-md-2"> 조회수 </th>
                <th class="col-md-10"> {board_view.bHit} </th>
            </tr>
            <tr class="row_g-2">
                <th class="col-md-2"> 이름 </th>
                <th class="col-md-10">
                    <input type="text" name="bName" class="form-control" value="{board_view.bName}">
                </th>
            </tr>
            <tr class="row_g-2">
                <th class="col-md-2"> 제목 </th>
                <th class="col-md-10">
                    <input type="text" name="bTitle" class="form-control" value="{board_view.bTitle}">
                </th>
            </tr>
            <tr class="row_g-2">
                <th class="col-md-2"> 내용 </th>
                <th class="col-md-10">
                    <textarea name="bContent" class="form-control" style="height: 400px">{board_view.bContent}</textarea>
                </th>
            </tr>
            <tr>
                <th class="col-md-12 text-end">
                    <input type="submit" value="수정" class="btn btn-primary"> &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;&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;&~
                    <a href="board" class="btn btn-primary">목록보기</a> &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;&~
                    <a href="delete?bId={board_view.bId}" class="btn btn-primary">삭제</a> &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~
                    <a href="board_reply?bId={board_view.bId}" class="btn btn-primary">답변</a>
                </th>
            </tr>
        </form>
    </table>
</div>
</section>
```

CONTENTS Board_Reply.jsp - 게시판 구현부분

```
<section style = "height:700px; width:1200px;">
<div class="container" style="position:relative; top: 40px; left:140px;">
  <table class="table" style="width:1200px;" >
    <form action="reply" method="post">
      <input type="hidden" name="bId" value="${board_reply_view.bId}">
      <input type="hidden" name="bGroup" value="${board_reply_view.bGroup}">
      <input type="hidden" name="bStep" value="${board_reply_view.bStep}">
      <input type="hidden" name="bIndent" value="${board_reply_view.bIndent}">
      <tr class="row g-2">
        <th class="col-md-2"> 번호 </th>
        <th class="col-md-10"> ${board_reply_view.bId} </th>
      </tr>
      <tr class="row g-2">
        <th class="col-md-2"> 조회수 </th>
        <th class="col-md-10"> ${board_reply_view.bHit} </th>
      </tr>
      <tr class="row g-2">
        <th class="col-md-2"> 이름 </th>
        <th class="col-md-10">
          <input type="text" name="bName" class="form-control" value="${board_reply_view.bName}">
        </th>
      </tr>
      <tr class="row g-2">
        <th class="col-md-2"> 제목 </th>
        <th class="col-md-10">
          <input type="text" name="bTitle" class="form-control" value="${board_reply_view.bTitle}">
        </th>
      </tr>
      <tr class="row g-2">
        <th class="col-md-2"> 내용 </th>
        <th class="col-md-10">
          <textarea name="bContent" class="form-control" style="height: 400px">${board_reply_view.bContent}</textarea>
        </th>
      </tr>
      <tr>
        <th class="col-md-12 text-end">
          <input type="submit" value="답변작성" class="btn btn-primary">
          <a href="board" class="btn btn-primary">목록</a>
        </th>
      </tr>
    </form>
  </table>
</div>
</section>
```

요청이 있을 시 페이지 연결부분

Controller

CONTENTS BController.java - MVC 부분

```
@Controller
public class BController {
    Service service;

    public JdbcTemplate template;

    @Autowired
    public void setTemplate(JdbcTemplate template) {
        this.template = template;
        Constant.template = this.template;
    }

    @RequestMapping("/board")
    public String board(Model model)
    {
        service = new BoardService();
        service.execute(model);

        return "board";
    }

    @RequestMapping("/board_write")
    public String board_write(Model model)
    {

        return "board_write";
    }

    @RequestMapping("write")
    public String write(HttpServletRequest request, Model model)
    {
        model.addAttribute("request", request);
        service = new WriteService();
        service.execute(model);

        return "redirect:board";
    }

    @RequestMapping("board_manage")
    public String board_manage(HttpServletRequest request, Model model)
    {
        model.addAttribute("request", request);
        service = new ContentService();
        service.execute(model);

        return "board_manage";
    }

    @RequestMapping(method=RequestMethod.POST, value="/modify")
    public String modify(HttpServletRequest request, Model model) {

        model.addAttribute("request", request);
        service = new ModifyService();
        service.execute(model);
        return "redirect:board";
    }
}
```

Controller-import부분

```
package com.mypage.lsh.controller;

import javax.servlet.http.HttpServletRequest;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.jdbc.core.JdbcTemplate;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;

import com.mypage.lsh.service.BoardService;
import com.mypage.lsh.service.ContentService;
import com.mypage.lsh.service.DeleteService;
import com.mypage.lsh.service.ModifyService;
import com.mypage.lsh.service.ReplyService;
import com.mypage.lsh.service.ReplyViewService;
import com.mypage.lsh.service.Service;
import com.mypage.lsh.service.WriteService;
import com.mypage.lsh.util.Constant;
```

```
@RequestMapping("/board_reply")
public String board_reply(HttpServletRequest request, Model model) {

    model.addAttribute("request", request);
    service = new ReplyViewService();
    service.execute(model);

    return "board_reply";
}

@RequestMapping("/reply")
public String reply(HttpServletRequest request, Model model) {

    model.addAttribute("request", request);
    service = new ReplyService();
    service.execute(model);

    return "redirect:board";
}

@RequestMapping("/delete")
public String delete(HttpServletRequest request, Model model) {

    model.addAttribute("request", request);
    service = new DeleteService();
    service.execute(model);

    return "redirect:board";
}
}
```

ORACLE 데이터와 연결되는 부분



DATA

CONTENTS BoardDto.java - MVC 부분

```
import java.sql.Timestamp;

public class BoardDto {
    int bId;
    String bName;
    String bTitle;
    String bContent;
    Timestamp bDate;
    int bHit;
    int bGroup;
    int bStep;
    int bIndent;

    public BoardDto() {

    }

    public BoardDto(int bId, String bName, String bTitle, String bContent, Timestamp bDate, int bHit, int bGroup, int bStep, int bIndent) {
        this.bId = bId;
        this.bName = bName;
        this.bTitle = bTitle;
        this.bContent = bContent;
        this.bDate = bDate;
        this.bHit = bHit;
        this.bGroup = bGroup;
        this.bStep = bStep;
        this.bIndent = bIndent;
    }

    public int getbId() {
        return bId;
    }

    public void setbId(int bId) {
        this.bId = bId;
    }

    public String getbName() {
        return bName;
    }

    public void setbName(String bName) {
        this.bName = bName;
    }

    public String getbTitle() {
        return bTitle;
    }

    public void setbTitle(String bTitle) {
        this.bTitle = bTitle;
    }

    public String getbContent() {
        return bContent;
    }

    public void setbContent(String bContent) {
        this.bContent = bContent;
    }

    public Timestamp getbDate() {
        return bDate;
    }

    public void setbDate(Timestamp bDate) {
        this.bDate = bDate;
    }

    public int getbHit() {
        return bHit;
    }

    public void setbHit(int bHit) {
        this.bHit = bHit;
    }

    public int getbGroup() {
        return bGroup;
    }

    public void setbGroup(int bGroup) {
        this.bGroup = bGroup;
    }

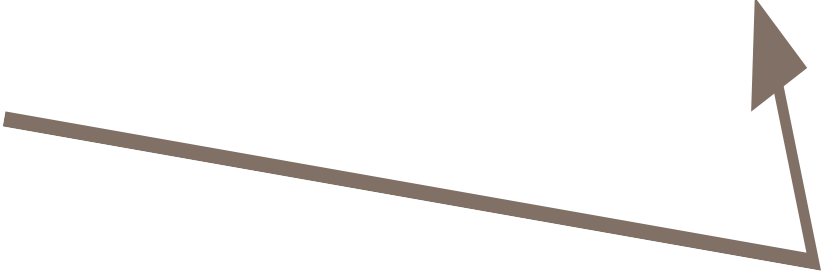
    public int getbStep() {
        return bStep;
    }

    public void setbStep(int bStep) {
        this.bStep = bStep;
    }

    public int getbIndent() {
        return bIndent;
    }

    public void setbIndent(int bIndent) {
        this.bIndent = bIndent;
    }
}
```

Dto 나머지 부분



CONTENTS BoardDao.java - Data 연결부분

```
public class BoardDao {

    DataSource datasource;

    JdbcTemplate template = null;

    public BoardDao() {
        template = Constant.template;
    }

    public ArrayList<BoardDto> board(){

        String query = "select bId,bName,bTitle,bContent,bDate,bHit,bGroup,bStep,bIndent from board order by bGroup desc,bStep asc";
        return (ArrayList<BoardDto>) template.query(query, new BeanPropertyRowMapper<BoardDto>(BoardDto.class));
    }

    public BoardDto board_view(String strID) {
        upHit(strID);
        String query = "select bId,bName,bTitle,bContent,bDate,bHit,bGroup,bStep,bIndent from board where bId = " + strID;
        return template.queryForObject(query, new BeanPropertyRowMapper<BoardDto>(BoardDto.class));
    }

    public void write(final String bName,final String bTitle,final String bContent) {
        template.update(new PreparedStatementCreator() {
            @Override
            public PreparedStatement createPreparedStatement(Connection con) throws SQLException {
                String query = "insert into board (bId,bName,bTitle,bContent,bHit,bGroup,bStep,bIndent) values (board_seq.nextval,?,?,?,0,board_seq.currval,0,0)";
                PreparedStatement pstmt = con.prepareStatement(query);
                pstmt.setString(1, bName);
                pstmt.setString(2, bTitle);
                pstmt.setString(3, bContent);

                return pstmt;
            }
        });
    }

    public void modify(String bId, String bName, String bTitle, String bContent) {
        String query = "update board set bName = ?, bTitle = ?, bContent = ? where bId = ?";
        template.update(query, new PreparedStatementSetter() {

            @Override
            public void setValues(PreparedStatement ps) throws SQLException {
                ps.setString(1, bName);
                ps.setString(2, bTitle);
                ps.setString(3, bContent);
                ps.setInt(4, Integer.parseInt(bId));
            }
        });
    }

    public BoardDto board_reply_view(final String strID) {
        upHit(strID);
        String query = "select bId,bName,bTitle,bContent,bDate,bHit,bGroup,bStep,bIndent from board where bId = " + strID;
        return template.queryForObject(query, new BeanPropertyRowMapper<BoardDto>(BoardDto.class));
    }
}
```

```
package com.mypage.lsh.boardDao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.util.ArrayList;

import javax.sql.DataSource;

import org.springframework.jdbc.core.BeanPropertyRowMapper;
import org.springframework.jdbc.core.JdbcTemplate;
import org.springframework.jdbc.core.PreparedStatementCreator;
import org.springframework.jdbc.core.PreparedStatementSetter;

import com.mypage.lsh.boardDto.BoardDto;
import com.mypage.lsh.utill.Constant;
```

CONTENTS BoardDao.java - Data 연결부분(2)

```
public void board_reply(final String bId,final String bName,final String bTitle,final String bContent, final String bGroup,final String bStep,final String bIndent) {
    board_reply_shape(bGroup,bStep);
    String query = "insert into board (bId, bName, bTitle, bContent, bGroup, bStep, bIndent) values (board_seq.nextval,?,?,?,?,?,?)";
    template.update(query, new PreparedStatementSetter() {

        @Override
        public void setValues(PreparedStatement ps) throws SQLException {
            ps.setString(1, bName);
            ps.setString(2, bTitle);
            ps.setString(3, bContent);
            ps.setInt(4, Integer.parseInt(bGroup));
            ps.setInt(5, Integer.parseInt(bStep)+1);
            ps.setInt(6, Integer.parseInt(bIndent)+1);
        }
    });
}

public void board_reply_shape(final String strGroup, final String strStep) {
    String query = "update board set bStep = bStep + 1 where bGroup = ? and bStep > ?";
    template.update(query, new PreparedStatementSetter() {

        @Override
        public void setValues(PreparedStatement ps) throws SQLException {
            ps.setInt(1, Integer.parseInt(strGroup));
            ps.setInt(2, Integer.parseInt(strStep));
        }
    });
}

public void delete(final String strID) {
    String query = "delete from board where bId = ?";
    template.update(query, new PreparedStatementSetter() {

        @Override
        public void setValues(PreparedStatement ps) throws SQLException {
            ps.setInt(1, Integer.parseInt(strID));
        }
    });
}

private void upHit(final String bId) {
    String query = "update board set bHit = bHit + 1 where bId = ?";

    template.update(query, new PreparedStatementSetter() {

        @Override
        public void setValues(PreparedStatement ps) throws SQLException {
            ps.setInt(1,Integer.parseInt(bId));
        }
    });
}
```

Windows 정품 인증

[설정]으로 이동하여 Windows를 정품 인증합니다

객체생성 부분

객체생성

CONTENTS Model 의존성주입 & Jdbc객체 - 의존성 주입(IOC)

```
package com.mypage.lsh.service;

import org.springframework.ui.Model;

public interface Service {
    public void execute(Model model);
}

package com.mypage.lsh.utill;

import org.springframework.jdbc.core.JdbcTemplate;

public class Constant {

    public static JdbcTemplate template;
}
```

CONTENTS Service 부분 - 객체생성부분

```
package com.mypage.lsh.service;

import java.util.ArrayList;

public class BoardService implements Service {

    @Override
    public void execute(Model model) {
        BoardDao dao = new BoardDao();
        ArrayList<BoardDto> dtos = dao.board();

        model.addAttribute("board", dtos);
    }
}
```

```
package com.mypage.lsh.service;

import java.util.Map;

public class ModifyService implements Service {

    @Override
    public void execute(Model model) {
        Map<String, Object> map = model.asMap();
        HttpServletRequest request = (HttpServletRequest) map.get("request");

        String bId = request.getParameter("bId");
        String bName = request.getParameter("bName");
        String bTitle = request.getParameter("bTitle");
        String bContent = request.getParameter("bContent");

        BoardDao dao = new BoardDao();
        dao.modify(bId, bName, bTitle, bContent);
    }
}
```

```
package com.mypage.lsh.service;

import java.util.Map;

public class ContentService implements Service {

    @Override
    public void execute(Model model) {
        Map<String, Object> map = model.asMap();
        HttpServletRequest request = (HttpServletRequest) map.get("request");
        String bId = request.getParameter("bId");

        BoardDao dao = new BoardDao();
        BoardDto dto = dao.board_view(bId);
        model.addAttribute("board_view", dto);
    }
}
```

```
package com.mypage.lsh.service;

import java.util.Map;

public class ReplyService implements Service {

    @Override
    public void execute(Model model) {
        Map<String, Object> map = model.asMap();
        HttpServletRequest request = (HttpServletRequest) map.get("request");

        String bId = request.getParameter("bId");
        String bName = request.getParameter("bName");
        String bTitle = request.getParameter("bTitle");
        String bContent = request.getParameter("bContent");
        String bGroup = request.getParameter("bGroup");
        String bStep = request.getParameter("bStep");
        String bIndent = request.getParameter("bIndent");

        BoardDao dao = new BoardDao();
        dao.board_reply(bId, bName, bTitle, bContent, bGroup, bStep, bIndent);
    }
}
```


CONTENTS Service 부분 - 객체생성부분(2)

```
package com.mypage.lsh.service;

import java.util.Map;

public class DeleteService implements Service {

    @Override
    public void execute(Model model) {
        Map<String, Object> map = model.asMap();
        HttpServletRequest request = (HttpServletRequest) map.get("request");
        String bId = request.getParameter("bId");

        BoardDao dao = new BoardDao();
        dao.delete(bId);
    }
}
```

```
package com.mypage.lsh.service;
```

```
import java.util.Map;
```

```
public class ReplyViewService implements Service {
```

```
    @Override
    public void execute(Model model) {
        Map<String, Object> map = model.asMap();
        HttpServletRequest request = (HttpServletRequest) map.get("request");

        String bId = request.getParameter("bId");

        BoardDao dao = new BoardDao();
        BoardDto dto = dao.board_reply_view(bId);

        model.addAttribute("board_reply_view",dto);
    }
}
```

```
package com.mypage.lsh.service;
```

```
import java.util.Map;
```

```
public class WriteService implements Service {
```

```
    @Override
    public void execute(Model model) {
        Map<String, Object> map = model.asMap();
        HttpServletRequest request = (HttpServletRequest) map.get("request");

        String bName = request.getParameter("bName");
        String bTitle = request.getParameter("bTitle");
        String bContent = request.getParameter("bContent");

        BoardDao dao = new BoardDao();
        dao.write(bName,bTitle,bContent);
    }
}
```

```
}
```

Q&A

→ 궁금한 점 있으신가요?

**Thank you
for
your attention!**

CONTACT



이메일

mysky0420@naver.com



홈페이지

MIRICAMPUS.com