

6장. SpringBoard 프로젝트



SpringBoard



메인 페이지입니다.

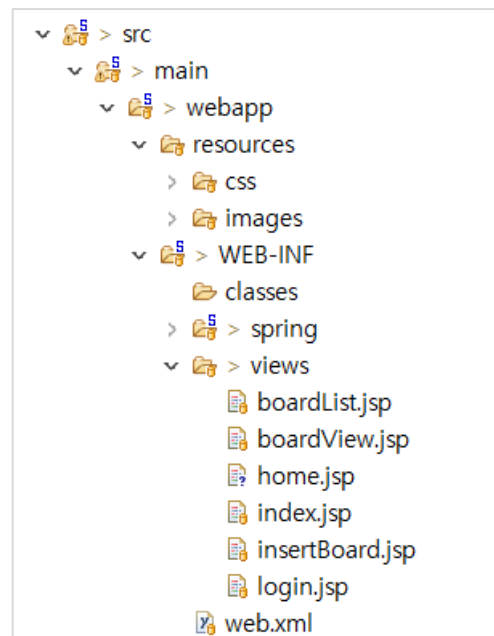
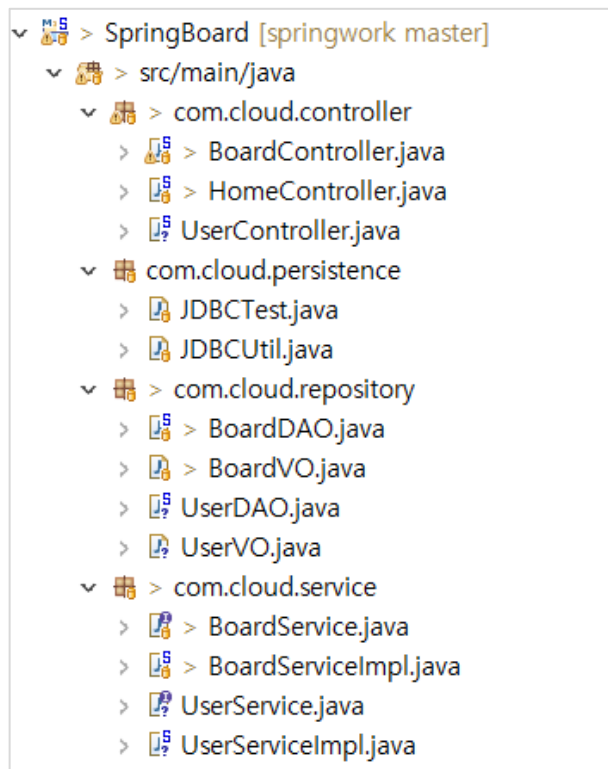


[로그인 바로가기](#)

[글 목록 바로가기](#)

SpringBoard

● 프로젝트 구조도



SpringBoard

● DB 구축

```
-- board 테이블 생성
CREATE TABLE board(
    bno NUMBER(5) PRIMARY KEY,
    title VARCHAR2(200),
    writer VARCHAR2(20),
    content VARCHAR2(2000),
    regdate DATE DEFAULT SYSDATE,
    cnt NUMBER(5) DEFAULT 0
);
-- 시퀀스 : 자동 순번
CREATE SEQUENCE seq;
```

```
-- 게시물 추가
INSERT INTO board(bno, title, writer, content)
VALUES (seq.NEXTVAL, '가입인사', '관리자', '잘 부탁드립니다...');
```

SQL | 인출된 모든 행: 1(0.06초)

BNO	TITLE	WRITER	CONTENT	REGDATE	CNT
1	가입인사	관리자	잘 부탁드립니다...	22/08/14	0



SpringBoard

● DB 구축

```
-- board 테이블 생성
CREATE TABLE board(
    bno NUMBER(5) PRIMARY KEY,
    title VARCHAR2(200),
    writer VARCHAR2(20),
    content VARCHAR2(2000),
    regdate DATE DEFAULT SYSDATE,
    cnt NUMBER(5) DEFAULT 0
);
-- 시퀀스 : 자동 순번
CREATE SEQUENCE seq;
```

```
-- 게시물 추가
INSERT INTO board(bno, title, writer, content)
VALUES (seq.NEXTVAL, '가입인사', '관리자', '잘 부탁드립니다...');
```

SQL | 인출된 모든 행: 1(0.06초)

BNO	TITLE	WRITER	CONTENT	REGDATE	CNT
1	가입인사	관리자	잘 부탁드립니다...	22/08/14	0



SpringBoard

- DB 구축

```
-- users 테이블 생성
CREATE TABLE users(
    id VARCHAR2(8) PRIMARY KEY,
    passwd VARCHAR2(8) NOT NULL,
    name VARCHAR2(20) NOT NULL,
    role VARCHAR2(5)
);
```

```
-- 회원 추가
INSERT INTO users VALUES ('test', 'test123', '관리자', 'Admin');
INSERT INTO users VALUES ('user1 ', 'user123', '장그래', 'User');
```

ID	PASSWD	NAME	ROLE
test	test123	관리자	Admin
user1	user123	장그래	User



SpringBoard

● DB 구축

```
-- board 테이블 생성
CREATE TABLE board(
    bno NUMBER(5) PRIMARY KEY,
    title VARCHAR2(200),
    writer VARCHAR2(20),
    content VARCHAR2(2000),
    regdate DATE DEFAULT SYSDATE,
    cnt NUMBER(5) DEFAULT 0
);
-- 시퀀스 : 자동 순번
CREATE SEQUENCE seq;
```

```
-- 게시물 추가
INSERT INTO board(bno, title, writer, content)
VALUES (seq.NEXTVAL, '가입인사', '관리자', '잘 부탁드립니다...');
```

SQL | 인출된 모든 행: 1(0.06초)

BNO	TITLE	WRITER	CONTENT	REGDATE	CNT
1	가입인사	관리자	잘 부탁드립니다...	22/08/14	0



- VO / DAO 만들기

```
@Setter
@Getter
public class BoardVO {
    private int bno;
    private String title;
    private String writer;
    private String content;
    @DateTimeFormat(pattern="yyyy-MM-dd")
    private Date regDate;
    private int cnt;
}
```



주소록(Address Book) 프로젝트

```
package com.common;
```

```
import java.sql.Connection;
```

JDBCUtil.java

```
public class JDBCUtil {
```

```
    private static String driverClass = "oracle.jdbc.OracleDriver";
```

```
    private static String url = "jdbc:oracle:thin:@localhost:1521:xe";
```

```
    private static String username = "c##jweb";
```

```
    private static String password = "54321";
```

```
    //DB 연결 메서드
```

```
    public static Connection getConnection() {
```

```
        try {
```

```
            Class.forName(driverClass);
```

```
            return DriverManager.getConnection(url, username, password);
```

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

```
            e.printStackTrace();
```

```
        }
```

```
        return null;
```

```
    }
```



주소록(Address Book) 프로젝트

//DB 연결 종료 메서드

```
public static void close(Connection conn, PreparedStatement pstmt) {  
    if(pstmt != null) {  
        try {  
            pstmt.close();  
        } catch (Exception e) {  
            e.printStackTrace();  
        }finally {  
            pstmt = null;  
        }  
    }  
  
    if(conn != null) {  
        try {  
            conn.close();  
        } catch (Exception e) {  
            e.printStackTrace();  
        }finally {  
            conn = null;  
        }  
    }  
}
```

JDBCUtil.java



회원 관리형 게시판 프로젝트

```
//연결 종료(ResultSet이 있는 경우)
public static void close(Connection conn, PreparedStatement pstmt, ResultSet rs) {
    if(rs != null) {
        try {
            rs.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    if(pstmt != null) {
        try {
            pstmt.close();
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            pstmt = null;
        }
    }
    if(conn != null) {
        try {
            conn.close();
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            conn = null;
        }
    }
}
```



- VO / DAO 만들기

```
@Repository
public class BoardDAO {

    private Connection conn = null;
    private PreparedStatement pstmt = null;
    private ResultSet rs = null;

    //SQL 쿼리 상수 선언
    private final String BOARD_INSERT =
        "INSERT INTO board(bno, title, writer, content) VALUES "
        + "(seq.NEXTVAL, ?, ?, ?)";
    private final String BOARD_LIST =
        "SELECT * FROM board ORDER BY bno DESC";
    private final String BOARD_ONE =
        "SELECT * FROM board WHERE bno = ?";
    private final String BOARD_UPDATE =
        "UPDATE board SET title = ?, content = ? WHERE bno = ?";
    private final String BOARD_DELETE =
        "DELETE FROM board WHERE bno = ?";
```



- VO / DAO 만들기

```
//글 등록
public void insertBoard(BoardVO vo) {
    System.out.println("==> insertBoard() 기능 처리");
    try {
        conn = JDBCUtil.getConnection();
        pstmt = conn.prepareStatement(BOARD_INSERT);
        pstmt.setString(1, vo.getTitle());
        pstmt.setString(2, vo.getWriter());
        pstmt.setString(3, vo.getContent());

        pstmt.executeUpdate();
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        JDBCUtil.close(conn, pstmt);
    }
}
```



SpringBoard

```
//글 목록
public List<BoardVO> getBoardList(){
    System.out.println("==> getBoardList() 기능 처리");
    List<BoardVO> boardList = new ArrayList<>();
    try {
        conn = JDBCUtil.getConnection();
        pstmt = conn.prepareStatement(BOARD_LIST);
        rs = pstmt.executeQuery();
        while(rs.next()) {
            BoardVO vo = new BoardVO();
            vo.setBno(rs.getInt("bno"));
            vo.setTitle(rs.getString("title"));
            vo.setWriter(rs.getString("writer"));
            vo.setContent(rs.getString("content"));
            vo.setRegDate(rs.getDate("regdate"));
            vo.setCnt(rs.getInt("cnt"));

            boardList.add(vo);
        }
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        JDBCUtil.close(conn, pstmt, rs);
    }
    return boardList;
}
```



SpringBoard

//글 상세 보기

```
public BoardVO getBoard(int bno) {  
    System.out.println("==> getBoard()");  
    BoardVO board = new BoardVO();  
    try {  
        conn = JDBCUtil.getConnection();  
        pstmt = conn.prepareStatement(BOARD_ONE);  
        pstmt.setInt(1, bno);  
        rs = pstmt.executeQuery();  
        if(rs.next()) {  
            board.setBno(rs.getInt("bno"));  
            board.setTitle(rs.getString("title"));  
            board.setWriter(rs.getString("writer"));  
            board.setContent(rs.getString("content"));  
            board.setRegDate(rs.getDate("regdate"));  
            board.setCnt(rs.getInt("cnt"));  
        }  
    } catch (SQLException e) {  
        e.printStackTrace();  
    } finally {  
        JDBCUtil.close(conn, pstmt, rs);  
    }  
    return board;  
}
```



SpringBoard

```
//글 수정하기
public void updateBoard(BoardVO vo) {
    System.out.println("==> updateBoard()");
    try {
        conn = JDBCUtil.getConnection();
        pstmt = conn.prepareStatement(BOARD_UPDATE);
        pstmt.setString(1, vo.getTitle());
        pstmt.setString(2, vo.getContent());
        pstmt.setInt(3, vo.getBno());

        pstmt.executeUpdate();
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        JDBCUtil.close(conn, pstmt);
    }
}
```



- VO / DAO 만들기

```
//글 삭제
public void deleteBoard(BoardVO vo) {
    System.out.println("==> deleteBoard()");
    try {
        conn = JDBCUtil.getConnection();
        pstmt = conn.prepareStatement(BOARD_DELETE);
        pstmt.setInt(1, vo.getBno());

        pstmt.executeUpdate();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
```



SpringBoard

```
//조회수
public void updateCount(int bno) {
    try {
        conn= JDBCUtil.getConnention();
        String sql = "SELECT cnt FROM board WHERE bno=?";
        pstmt = conn.prepareStatement(sql);
        pstmt.setInt(1, bno);
        rs = pstmt.executeQuery();
        int cnt = 0;
        if(rs.next()) {
            cnt = rs.getInt("cnt") + 1;
        }

        //조회수 update 쿼리
        sql = "UPDATE board SET cnt = ? WHERE bno=?";
        pstmt = conn.prepareStatement(sql);
        pstmt.setInt(1, cnt);
        pstmt.setInt(2, bno);
        pstmt.executeUpdate();
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        JDBCUtil.close(conn, pstmt, rs);
    }
}
```



- Service 계층

```
public interface BoardService {  
  
    public void insert(BoardVO vo);    //글 쓰기  
  
    public List<BoardVO> getBoardList(); //글 목록  
  
    public BoardVO getBoard(int bno); //글 상세보기  
  
    public void delete(BoardVO vo); //글 삭제  
  
    public void update(BoardVO vo); //글 수정  
  
    public void updateCount(int bno); //조회수  
  
}
```



- Service 계층

```
@Service("boardService")
public class BoardServiceImpl implements BoardService{

    @Autowired
    private BoardDAO dao;

    @Override
    public void insert(BoardVO vo) {    //글쓰기
        dao.insertBoard(vo);
    }

    @Override
    public List<BoardVO> getBoardList() {    //글 목록
        return dao.getBoardList();
    }

    @Override
    public BoardVO getBoard(int bno) {    //상세 보기
        return dao.getBoard(bno);
    }
}
```



- Service 계층

```
@Override
public void delete(BoardVO vo) { //삭제
    dao.deleteBoard(vo);
}

@Override
public void update(BoardVO vo) { //수정
    dao.updateBoard(vo);
}

@Override
public void updateCount(int bno) { //조회수
    dao.updateCount(bno);
}
}
```



- Controller 계층

```
@Controller
public class HomeController {

    private static final Logger logger = LoggerFactory.getLogger(HomeController.class);

    @RequestMapping(value = "/", method = RequestMethod.GET)
    public String home(Locale locale, Model model) {
        logger.info("Welcome home! The client locale is {}.", locale);

        Date date = new Date();
        DateFormat dateFormat = DateFormat.getDateInstance(DateFormat.LONG);

        String formattedDate = dateFormat.format(date);

        model.addAttribute("serverTime", formattedDate );

        return "index";
    }
}
```



- Controller 계층

```
@Controller
public class BoardController {

    @Autowired
    private BoardService service;

    @RequestMapping("/boardList")
    public String getBoardList(Model model) { //게시글 목록 요청
        List<BoardVO> boardList = service.getBoardList();
        model.addAttribute("boardList", boardList); //model-"boardList"
        return "boardList";
    }

    @RequestMapping("/boardView")
    public String getBoard(int bno, Model model) { //상세 보기 요청
        service.updateCount(bno); //조회수 증가
        BoardVO board = service.getBoard(bno); //상세 보기 처리
        model.addAttribute("board", board); //model-"board"
        return "boardView";
    }
}
```



SpringBoard

```
@RequestMapping(value="/insertBoard", method=RequestMethod.GET)
public String insertBoard() { //글쓰기 폼 페이지 요청
    return "insertBoard";
}

/*@RequestMapping(value="/insertBoard", method=RequestMethod.POST)
public String insertBoard(HttpServletRequest request) throws Unsupported
    //사용자 입력 정보 추출
    request.setCharacterEncoding("utf-8");

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

    //db 연동
    BoardVO vo = new BoardVO();
    vo.setTitle(title);
    vo.setWriter(writer);
    vo.setContent(content);

    boardService.insert(vo);
    return "redirect:boardList";
}*/
```



- Controller 계층

```
@RequestMapping(value="/insertBoard", method=RequestMethod.POST)  
public String insertBoard(BoardVO vo){ //글쓰기 처리  
    //command 객체 - BoardVO -> HttpServletRequest request 대체함  
    service.insert(vo);  
    return "redirect:boardList";  
}  
  
@GetMapping("/deleteBoard")  
public String deleteBoard(BoardVO vo) { //글 삭제 요청  
    service.delete(vo);  
    return "redirect:boardList";  
}  
  
@PostMapping("/updateBoard")  
public String updateBoard(BoardVO vo) { //글 수정 요청  
    service.update(vo);  
    return "redirect:boardList";  
}  
}
```



index.jsp

```
<title>Welcome~</title>
<link rel="stylesheet" href="/resources/css/style.css">
</head>
<body>
  <div id="container">
    <section id="main">
      <h2>메인 페이지입니다.</h2>
      <hr>
      
      <h4>
        <a href="/login">로그인 바로가기</a>&nbsp; &nbsp; &nbsp;
        <a href="/boardList">글 목록 바로가기</a>
      </h4>
    </section>
  </div>
</body>
```



글 목록

[Log-out](#)

번호	제목	작성자	등록일	조회수
8	공지사항	관리자	2022-08-25 05:26:19	0
7	여름은 가고	한여름	2022-08-25 05:07:57	1
5	환절기 감기 조심	손흥민	2022-08-25 04:45:23	0
4	test - 수정후	관리자	2022-08-25 04:06:51	5
2	귀뚜라미	김가을	2022-08-25 03:59:38	2
1	가입인사	관리자	2022-08-25 03:30:44	0

글쓰기



SpringBoard

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>글 목록</title>
<link rel="stylesheet" href="/resources/css/style.css">
</head>
<body>
    <div id="container">
        <section id="list">
            <h2>글 목록</h2>
            <h3>
                <a href="/logout">Log-out</a>
            </h3>
            <table class="tbl_list">
                <tr>
                    <th>번호</th>
                    <th>제목</th>
                    <th>작성자</th>
                    <th>등록일</th>
                    <th>조회수</th>
                </tr>
```

boardList.jsp



```
<c:forEach var="board" items="${boardList}">
<tr>
  <td>${board.bno}</td>
  <td><a href="/boardView?bno=${board.bno}">${board.title}</a></td>
  <td>${board.writer}</td>
  <td><fmt:formatDate value="${board.regDate}" pattern="yyyy-MM-dd hh:mm:ss"/>
  <td>${board.cnt}</td>
</tr>
</c:forEach>
</table>
<div class="btn_box">
  <a href="./insertBoard"><input type="button" value="글쓰기"></a>
</div>
```



남 이 남

제목	공지 사항
작성자	관리자
내용	<div> <div>서비스 점검</div> <div>일시 : 2022. 9. 9</div> </div>

등록

목록

- 한글 인코딩 처리(Filter) - web.xml

```
<!-- 한글 인코딩 필터 -->
<filter>
  <filter-name>encoding</filter-name>
  <filter-class>
    org.springframework.web.filter.CharacterEncodingFilter
  </filter-class>
  <init-param>
    <param-name>encoding</param-name>
    <param-value>UTF-8</param-value>
  </init-param>
</filter>

<filter-mapping>
  <filter-name>encoding</filter-name>
  <servlet-name>appServlet</servlet-name>
</filter-mapping>
```



SpringBoard

```
<section id="register">
  <h2>글 등록</h2>
  <hr>
  <form action="/insertBoard" method="post">
    <table class="tbl_reg">
      <tr>
        <td width="100">제목</td>
        <td align="left"><input type="text" name="title" size="50"></td>
      </tr>
      <tr>
        <td>작성자</td>
        <td><input type="text" name="writer"></td>
      </tr>
      <tr>
        <td>내용</td>
        <td><textarea name="content" cols="50" rows="10"></textarea></td>
      </tr>
      <tr>
        <td colspan="2" align="center">
          <input type="submit" value="등록">
          <a href="/boardList"><input type="button" value="목록"></a>
        </td>
      </tr>
    </table>
  </form>
</section>
```

insertBoard.jsp



글 상세 보기

[Log-out](#)

boardView.jsp

제목	<input type="text" value="test - 수정후"/>
작성자	<input type="text" value="관리자"/>
내용	<div>test --> <u>수정 후</u></div>
등록일	<input type="text" value="2022-08-25"/>
조회수	<input type="text" value="0"/>
<div><input type="button" value="글 수정"/> <input type="button" value="삭제"/> <input type="button" value="목록"/></div>	



SpringBoard

boardView.jsp

```
<section id="list">
  <h2>글 상세 보기</h2>
  <h3>
    <a href="/logout">Log-out</a>
  </h3>
  <hr>
  <form action="/updateBoard" method="post">
  <input type="hidden" name="bno" value="${board.bno}">
    <table class="tbl_view">
      <tr>
        <td width="100">제목</td>
        <td align="left"><input type="text" name="title" value="${board.title}"></td>
      </tr>
      <tr>
        <td>작성자</td>
        <td><input type="text" name="writer" value="${board.writer}"></td>
      </tr>
      <tr>
        <td>내용</td>
        <td><textarea name="content" cols="62" rows="10">${board.content}</textarea></td>
      </tr>
    </table>
  </form>
</section>
```



```
<tr>
  <td>등록일</td>
  <td><input type="text" name="regDate" value="${board.regDate}"></td>
</tr>
<tr>
  <td>조회수</td>
  <td><input type="text" name="cnt" value="${board.cnt}"></td>
</tr>
<tr>
  <td colspan="2" align="center">
    <input type="submit" value="글 수정">
    <a href="/deleteBoard?bno=${board.bno}"
      onclick="return confirm('정말로 삭제하시겠습니까?')">
      <input type="button" value="삭제"></a>
    <a href="/boardList"><input type="button" value="목록"></a>
  </td>
</tr>
</table>
</form>
</section>
```



로그인

아이디	<input type="text"/>
비밀번호	<input type="password"/>
<input type="button" value="로그인"/>	



글 목록

test님 환영합니다... [Logout](#)

번호	제목	작성자	등록일	조회수
8	공지사항	관리자	2022-08-25 05:26:19	0
7	여름은 가고	한여름	2022-08-25 05:07:57	1
5	환절기 감기 조심	손흥민	2022-08-25 04:45:23	0
4	test - 수정후	관리자	2022-08-25 04:06:51	5
2	귀뚜라미	김가을	2022-08-25 03:59:38	2
1	가입인사	관리자	2022-08-25 03:30:44	0

글쓰기



- VO / DAO 만들기

```
@Getter  
@Setter  
public class UserVO {  
    private String id;  
    private String passwd;  
    private String name;  
    private String role;  
}
```



- VO / DAO 만들기

```
@Repository
public class UserDao {
    private Connection conn = null;
    private PreparedStatement pstmt = null;
    private ResultSet rs = null;

    private final String LOGIN =
        "SELECT * FROM users WHERE id = ? and passwd = ?";

    //로그인 처리
    public boolean login(UserVO vo) {
        try {
            conn = JDBCUtil.getConnection();
            pstmt = conn.prepareStatement(LOGIN);
            pstmt.setString(1, vo.getId());
            pstmt.setString(2, vo.getPasswd());
            rs = pstmt.executeQuery();
            if(rs.next()) {
                return true;
            }
        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            JDBCUtil.close(conn, pstmt, rs);
        }
        return false;
    }
}
```



- VO / DAO 만들기

```
@Repository
public class UserDao {
    private Connection conn = null;
    private PreparedStatement pstmt = null;
    private ResultSet rs = null;

    private final String LOGIN =
        "SELECT * FROM users WHERE id = ? and passwd = ?";

    //로그인 처리
    public boolean login(UserVO vo) {
        try {
            conn = JDBCUtil.getConnection();
            pstmt = conn.prepareStatement(LOGIN);
            pstmt.setString(1, vo.getId());
            pstmt.setString(2, vo.getPasswd());
            rs = pstmt.executeQuery();
            if(rs.next()) {
                return true;
            }
        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            JDBCUtil.close(conn, pstmt, rs);
        }
        return false;
    }
}
```



- Service 계층

```
import com.cloud.repository.UserVO;  
  
public interface UserService {  
    public boolean login(UserVO vo); //로그인 처리  
}
```



SpringBoard

- Service 계층

```
@Service
public class UserServiceImpl implements UserService{

    @Autowired
    private UserDao dao;

    @Override
    public boolean login(UserVO vo) {
        return dao.login(vo);
    }
}
```



● Controller 계층

```
@Controller
public class UserController{

    @Autowired
    private UserService service;

    @RequestMapping(value="/login", method=RequestMethod.GET)
    public String loginView() { //로그인 폼 페이지 요청
        return "login";
    }

    @RequestMapping(value="/login", method=RequestMethod.POST)
    public String login(UserVO vo, HttpSession session) { //로그인 처리 요청
        boolean result = service.login(vo);

        if(result) {
            session.setAttribute("sessionId", vo.getId()); //세션 발급
            return "redirect:boardList";
        }else {
            return "login";
        }
    }
}
```



- Controller 계층

```
@GetMapping("/logout")
public String logout(HttpSession session) { //로그 아웃 처리 요청
    session.invalidate(); //세션 삭제
    return "index";
}
}
```



```
<title>로그인</title>
<link rel="stylesheet" href="/resources/css/style.css">
<script>
    function checkForm(){
        let form = document.login

        if(form.id.value == ""){
            alert("아이디를 입력해주세요");
            form.id.focus();
            return false;
        }
        if(form.passwd.value == ""){
            alert("비밀번호를 입력해주세요");
            form.passwd.focus();
            return false;
        }
    }
</script>
</head>
```

login.jsp



login.jsp

```
<div id="container">
  <section id="login">
    <h2>로그인</h2>
    <form action="/login" method="post"
      name="login" onsubmit="return checkForm()">
      <table class="tbl_login">
        <tr>
          <td>아이디</td>
          <td><input type="text" name="id"></td>
        </tr>
        <tr>
          <td>비밀번호</td>
          <td><input type="password" name="passwd"></td>
        </tr>
        <tr>
          <td colspan="2">
            <button type="submit">로그인</button>
          </td>
        </tr>
      </table>
    </form>
  </section>
</div>
```



boardList.jsp

[illegible]

로그인

아이디나 비밀번호를 확인해주세요.

아이디	<input type="text"/>
비밀번호	<input type="password"/>
<input type="button" value="로그인"/>	

SpringBoard

```
<section id="login">
  <h2>로그인</h2>
  <c:if test="${error==1}">
    <p class="error"><c:out value="아이디나 비밀번호를 확인해주세요." /></p>
  </c:if>
  <form action="/login" method="post">
    <table class="tbl_login">
      <tr>
```

login.jsp

```
@RequestMapping(value="/login", method=RequestMethod.POST)
public String login(UserVO vo, HttpSession session, Model model) { //로그인
    boolean result = service.login(vo);
    System.out.println(result);
    if(result==true) {
        session.setAttribute("sessionId", vo.getId()); //세션 발급
        return "redirect:boardList";
    }else {
        int error = 1;
        model.addAttribute("error", error);
        return "login";
    }
}
```

userController.java

