

240223 스프링 프레임워크 5일차 :: 추르 없는 집

BoardController

```
@PostMapping("/delete")
```

```
public String delete(Model model, String pass, int no,  
PrintWriter out, HttpServletResponse response) {
```

```
// 비밀번호가 맞는지 체크
```

```
boolean result = service.isPassCheck(no, pass);
```

```
// 비밀번호가 맞지 않으면 - 사용자에게 알림을 응답하고 프로그램 종료
```

```
if(!result) {
```

```
// 응답을 직접 생성
```

```
response.setContentType("text/html; charset=utf-8");
```

```
out.println("<script>");
```

```
out.println(" alert('비밀번호가 맞지 않음');");
```

```
out.println(" history.back();");
```

```
out.println("</script>");
```

```
return null;
```

```
}
```

```
// 비밀번호가 맞으면 게시 글을 삭제
```

```
service.deleteBoard(no);
```

```
// 게시 글 삭제가 완료되면 게시 글 리스트로 redirect
return "redirect:boardList";
}
```

```
@PostMapping("/updateProcess")
```

```
public String update(Model model, Board board,
    PrintWriter out, HttpServletResponse response) {
```

```
// 비밀번호가 맞는지 체크
```

```
boolean result = service.isPassCheck(board.getNo(), board.getPass());
```

```
// 비밀번호가 맞지 않으면 - 사용자에게 알림을 응답하고 프로그램 종료
```

```
if(!result) {
```

```
// 응답을 직접 생성
```

```
response.setContentType("text/html; charset=utf-8");
```

```
out.println("<script>");
```

```
out.println(" alert('비밀번호가 맞지 않음');");
```

```
out.println(" history.back();");
```

```
out.println("</script>");
```

```
return null;
```

```
}
```

```
// 게시 글을 수정
```

```
service.updateBoard(board);
```

```
// 게시 글 수정이 완료되면 redirect
return "redirect:boardList";
}
```

```
@PostMapping("/update")
```

```
public String update(Model model, String pass, int no,
    PrintWriter out, HttpServletResponse response) {
```

```
// 비밀번호가 맞는지 체크
```

```
boolean result = service.isPassCheck(no, pass);
```

```
// 비밀번호가 맞지 않으면 - 사용자에게 알림을 응답하고 프로그램 종료
```

```
if(!result) {
```

```
// 응답을 직접 생성
```

```
response.setContentType("text/html; charset=utf-8");
```

```
out.println("<script>");
```

```
out.println(" alert('비밀번호가 맞지 않음');");
```

```
out.println(" history.back();");
```

```
out.println("</script>");
```

```
return null;
```

```
}
```

```
// 비밀번호가 맞으면 - no에 해당하는 게시 글 정보를 읽어와 모델에 저장
```

```
model.addAttribute("board", service.getBoard(no));
```

```
return "/updateForm";
```

```
}
```

```
//@RequestMapping("/writeProcess")
// 파라미터를 받아서 도메인 객체로 넘겨주는 커멘트 객체
@PostMapping("/writeProcess")
public String insertBoard(Board board) {
    System.out.println("before - no : " + board.getNo());
    // 입력 - Board 객체로 만들어서
    // 게시 글 등록 => service
    service.insertBoard(board);

    System.out.println("after - no : " + board.getNo());
    // 최종적으로 응답
    return "redirect:boardList";
}
```

BoardDao

```
public interface BoardDao {

    // no에 해당하는 비밀번호를 db 테이블에서 읽어오는 메서드
    public abstract String isPassCheck(int no);
    public abstract List<Board> boardList();
    Board getBoard(int no);
    void insertBoard(Board board);
    void updateBoard(Board board);
    void deleteBoard(int no);
}
```

BoardDaoImpl

```
public void deleteBoard(int no) {  
    sqlSession.delete(NAME_SPACE + ".deleteBoard", no);  
}
```

```
public void updateBoard(Board board) {  
    sqlSession.update(NAME_SPACE + ".updateBoard", board);  
}
```

```
public String isPassCheck(int no) {  
    return sqlSession.selectOne(NAME_SPACE + ".isPassCheck", no);  
}
```

BoardService

// 게시 글 삭제하기

```
void deleteBoard(int no);
```

// 게시 글 수정하기

```
void updateBoard(Board board);
```

// 게시 글 수정, 삭제에서 비밀번호 체크

```
boolean isPassCheck(int no, String pass);
```

BoardServiceImpl

// 게시 글 삭제하기

```
public void deleteBoard(int no) {  
    boardDao.deleteBoard(no);  
}
```

// 게시 글 수정하기

```
public void updateBoard(Board board) {  
    boardDao.updateBoard(board);  
}
```

// 게시 글 수정, 삭제에서 비밀번호 체크

```
public boolean isPassCheck(int no, String pass) {  
    boolean result = false;
```

```
    String dbPass = boardDao.isPassCheck(no);  
    if(dbPass.equals(pass)) {  
        result = true;  
    }
```

```
    return result;  
}
```

BoardMapper

```
<delete id="deleteBoard">
```

```
DELETE FROM springbbs
```

```
WHERE no=#{no}
```

```
</delete>
```

```
<update id="updateBoard" parameterType="Board">
```

```
UPDATE springbbs
```

```
SET title=#{title}, content=#{content}, reg_date=SYSDATE()
```

```
WHERE no=#{no}
```

</update>

<select id="isPassCheck" resultType="String">

SELECT

pass

FROM springbbs

WHERE no=#{no}

</select>

formCheck.js

```
$("#updateForm").on("submit", function() {
```

```
if($("#writer").val().length <= 0) {
```

```
alert("작성자가 입력되지 않음");
```

```
$("#writer").focus();
```

```
return false;
```

```
}
```

```
if($("#title").val().length <= 0) {
```

```
alert("제목이 입력되지 않음");
```

```
$("#title").focus();
```

```
return false;
```

```
}
```

```
if($("#pass").val().length <= 0) {
```

```
alert("비밀번호가 입력되지 않음");
```

```
$("#pass").focus();
```

```
return false;
```

```
}
```

```
if($("#content").val().length <= 0) {
```

```
alert("내용이 입력되지 않음");  
$("#content").focus();  
return false;  
}
```

```
});
```

```
$("#detailDelete").on("click", function() {  
let pass = $("#pass").val();  
if(pass.length <= 0) {  
alert("게시 글을 삭제하려면 비밀번호를 입력해 주세요");  
return false; // 기본 이벤트와 이벤트 전파를 취소  
}  
}
```

```
$("#rPass").val(pass);  
$("#checkForm").attr("action", "delete");  
$("#checkForm").attr("method", "post");  
$("#checkForm").submit();  
  
});
```

```
$("#detailUpdate").on("click", function() {  
let pass = $("#pass").val();  
if(pass.length <= 0) {  
alert("비밀번호를 입력해 주세요");  
}
```



```
return false; // 기본 이벤트와 이벤트 전파를 취소
}
```

```
$("#rPass").val(pass);
$("#checkForm").attr("action", "update");
$("#checkForm").attr("method", "post");
$("#checkForm").submit();

});
```

boardDetail

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
<div class="row my-5" id="global-content">
<div class="col-10 offset-1">
<form name="checkForm" id="checkForm">
<input type="hidden" name="no" id="no" value="${board.no}">
<input type="hidden" name="pass" id="rPass">
</form>
```

updateForm

클리어하기