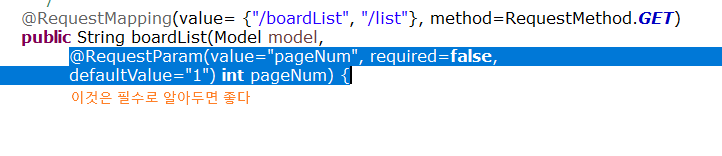
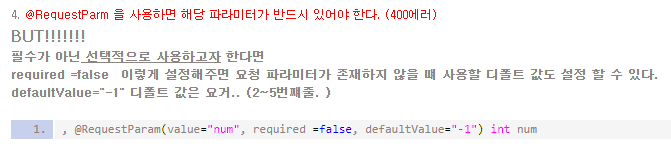
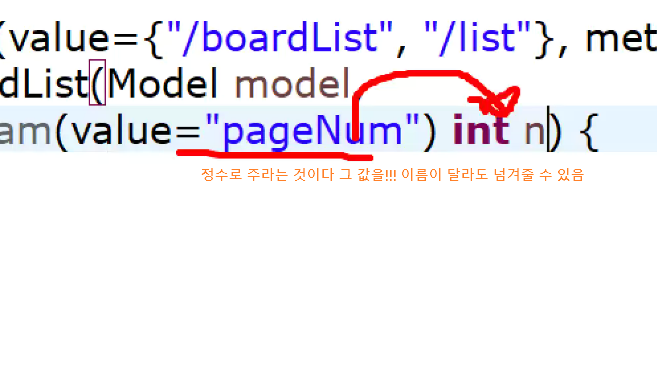
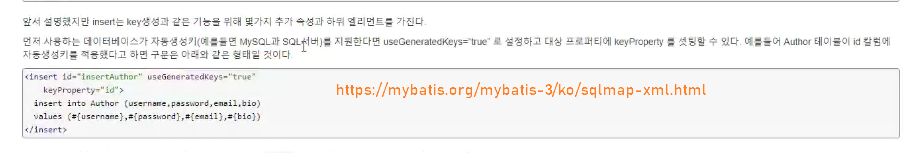
2024-02-27

* 게시글 수정, 삭제 , 페이징 처리 진도 나감 (~p70)

  
아래와 같은 설명처럼 기본값을 false로 로 주고 파라미터 존재하지 않을 때 1로 준다는 것

  
<https://blog.naver.com/conviction20/220549251602>

  
그런데 우리는 이름을 같게 주었다 -> (value=”pagaNum”) int pageNum) {



**--- BoardMapper.xml**

mybatis가 useGeneratedKeys ="true" deyProperty="no"를 읽어들인다

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*수정폼 p46\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

post방식으로 보내는 것이 필요함

수정하기 필요한것

- 비밀번호 값이 같은지 확인해야함 (글 썼는지 판단하는) -> boardDetail.jsp 파일에서 hidden 타입의 input을 만들어서 값이 잘 전달되는지 확인했다

- 수정하기 버튼이 클릭되었을 때 값을 읽어와서 같은지 판단하겠다 -> post방식인지 boardDetail.jsp에서 해도 되고 formcheck.js에서 줘도 된다

**--- formcheck.js**

attribute가 js에서는 attr이다

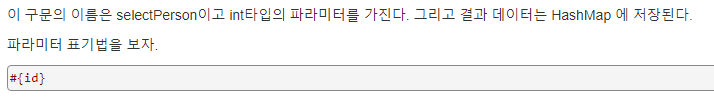
submit() -> 이벤트 강제로 발생시키는 것

여기에 써준 acting , update가 Controller.java에서 정의해준 것

\*

BoardController -> BoardService -> BoardDao(isPassCheck 호출 파라미터는 no로 지정) -> BoardMapper.xml

+++++++++++++왜 #{} 이렇게 써주는 걸까? => 파라미터를 이렇게 써주기로 정의해서!!

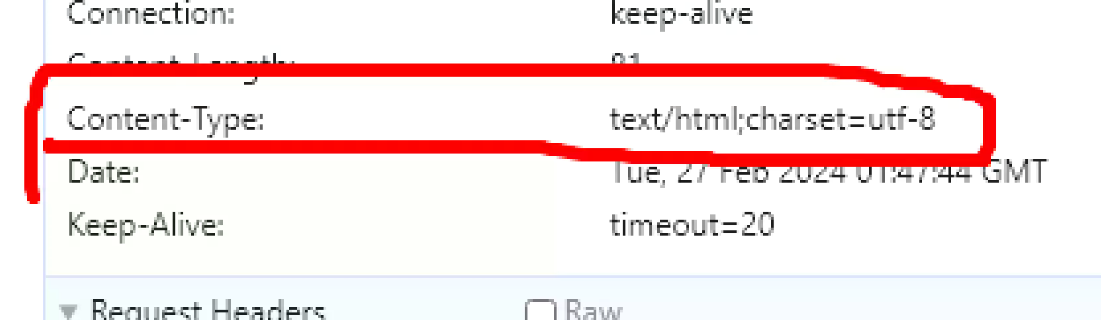


Dao 에서 no를 받고 Service에서 비밀번호 맞는지 ㅊ ㅔ크 -> controller에서 그 값을 가지고 와서 맞으면 어떻게 할건지 맞지않으면 어떻게 할건지 입력

**--- boardController.java**

// 비밀번호가 맞는지 체크 isPass에서 true와 false의 값을 가지고 와서 if문으로 true일 때, false일 때를 맞춰줌

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

content type을 제대로 지정해줘야함 -> response.setContentType("text/html; charset=utf-8"); -> 그 위에 httpServletResponse response 지정해줌  


**--- updateForm.jsp**

게시글 쓰기는 전달만 해주면 되는데 수정하기는 모델에 담음 것을 뿌려줘야함

no를 숨겨놓고 보내줘야함 -> <input type="hidden" name="no" value="${board.no}">

글쓴이 readonly 주면은 수정할 수 없게 고정 가능

++++++++++++++??? 어디로 리턴되는지 어떻게 아는건가?

return 하면 servlet content 에서 확인 -> 저기에서 view 페이지 조합해서 만드는 것이고

return 하면 자동적으로 확인해서 그 값을 확인한다 디스패쳐에서 확인함

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*삭제하기\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

delete도 post 방식

(웹페이지에서 작동안되면 ctrl+shift+r 해주기)

DELETE FROM springbbs WHERE no=#{no}

**--- BoardDao** 에서 정의 해주고 BoardDaoImpl에서 파라미터 있는 것으로 sqlSession.delete 해준다

**--- BoardService.java**

boardDao를 가지고 deleteBoard(no);를 해준다. //게시글 삭제하기

**--- BoardController.java**

게시글 삭제 하고 삭제 완료되면 게시글 리스트로 redirect 해준다

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*페이징 처리(p58)\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

게시 글 리스트 페이지에서 한 페이지에 게시 글을 몇개씩 출력할지를 정해서   
그 개수만큼씩만 한 페이지에 출력되도록 페이징하기

1페이지에 10개

0번부터 10개 가지고 와

오라클은 중첩해서 가지고 와야하는데 MYSql은 간단하게 limit 0, 10; 0부터 10개를 바로 쓸 수 있다

**--- BoardMapper.xml**

boardResult를 사용하지 않으면 select no, title, writer, content, reg\_date as redDate, 등 이렇게 다 써줘야 한다-> DB이름을 spring에서 지정한 이름으로 바꿔주는 것이 AS 인데 생략도 가능.

parameterType을 hashmap으로 지정해줌!

★ parameterType은 생략가능하지만 resultType은 생략 불가능!! 생략하면 오류날것

**--- BoardDaoImpl.java**

파라미터 2개를 던져야 한다 (리스트, 맵 있다 대표적으로 리스트는 순서가 있어서 구분할 수가 없다 -> Map으로 넣어주기)

Map<String, Object> param = new HashMap<>();

param.put("startRow", startRow);

-> 각각 데이터 넣어줌

**--- BoardServiceImpl.java**

배열로 넘겨줄수 있냔느 거다

한개가 아니라 여러개 담을 수 있는 Map으로 바꿔줄 것이다

List -> Map으로 바뀜

// 한 페이지에 출력할 게시 글의 수

// 한 페이지에 보여질 페이지 그룹 수

Service에서 정의해 놓은 boardList를 Override해서 재정의하기

pagenum을 //비지니스 로직 처리 - 페이징 처리

// 현재 페이지의 시작행, 페이지 사이즈가 들어가야 한다

// 현재 페이지의 시작행 -> startRow

나머지가 존재하면 한페이지가 더 추가하도록 삼항연산자를 통해 입력해줌

1페이지도 1 10페이지도 1이되어야 하니까

// 10,20 페이지일 때 딱 떨어지는 경우에만 PAGE\_GROUP 만큼 빼주면 된다

-> 삼항연산자로 나눠서 0이 되는 경우 PAGE\_GROUP이고, 아니면 0으로 해주기

int startPage = (currentPage / PAGE\_GROUP) \* PAGE\_GROUP + 1

- (currentPage % PAGE\_GROUP == 0 ? PAGE\_GROUP : 0);

// 뷰에 출력할 요청 처리 결과 데이터 - 모델을 Map에 저장

++++++++++++++++++??? Map의 구조는 무엇인가?





List로 해주면 순서가 있어서 구분할 수 없기 때문에 Map으로 모델을 지정한 것 (위의 내용)

**--- BoardController.java**

@RequestParam(value="pageNum", required=false, defaultValue="1") int pageNum) {

// 기본값이 false 존재하지 않을 때 기본 값을 1로 정해준 것

부트스트랩 페이지네이션 부분 참고해서 봤다 (사이트)

**--- boardList.jsp**

현재 페이지는 링크하지 않고 active 시킨다

현재 페이지가 아닌 페이지는 링크

++++++++++++++++왜 EL로 감싸줘야 하는가?

startPage 가 숫자인가? -> 서비스에서 넣어준 것 숫자맞지? 그러면 숫자여야 한다 -> **EL로 감싸줘야 숫자가 된다 그래야 모델데이터가 숫자로 들어온다** 그런데 EL로 감싸주지 않으면 그냥 문자열 자체의 startPage라는 것이다 -> 오류남

**게시 글 상세보기 요청처리 전 까지 진도 나감(~70)**