

Mybatis 게시판 제작(with paging, search)2

프로젝트명 : B07dMybatisBoard

수정하기

컨트롤러 : com.edu.springboot.MainController.java

```
07 @GetMapping("/edit.do")
08 public String boardEditGet(Model model, BoardDTO boardDTO) {
09     boardDTO = dao.view(boardDTO);
10     model.addAttribute("boardDTO", boardDTO);
11     return "edit";
12 }
13 @PostMapping("/edit.do")
14 public String boardEditPost(BoardDTO boardDTO) {
15     int result = dao.edit(boardDTO);
16     System.out.println("글수정결과:" + result);
17     return "redirect:view.do?idx="+ boardDTO.getIdx();
18 }
```

인터페이스 : com.edu.springboot.jdbc.IBoardService.java

```
8 @Mapper
9 public interface IBoardService {
10     //Paging처리를 위해 게시물의 갯수를 카운트
11     public int getTotalCount(ParameterDTO parameterDTO);
12     //게시판 목록(페이징 기능 추가)
13     public ArrayList<BoardDTO> listPage(ParameterDTO parameterDTO);
14     //작성하기
15     public int write(@Param("_name") String name,
16                     @Param("_title") String title,
17                     @Param("_content") String content);
18     //내용보기
19     public BoardDTO view(BoardDTO boardDTO);
20     //수정하기
21     public int edit(BoardDTO boardDTO);
22     //삭제하기
23     public int delete(String idx);
24 }
```

매퍼 : resources/mapper/BoardDAO.xml

```
63 <update id="edit" parameterType="com.edu.springboot.jdbc.BoardDTO">
64     UPDATE myboard SET
65     name=#{name}, title=#{title}, content=#{content}
66     WHERE idx=#{idx}
67 </update>
```

뷰 : webapp/WEB-INF/views/edit.jsp

```
<body>
    <h2>게시판 수정(Mybatis)</h2>
    <form name="writeFrm" method="post"
        action="/edit.do" onsubmit="return validateForm(this);">
    <input type="hid-den" name="idx" value="${boardDTO.idx}" />
    <table border="1" width="90%">
        <tr>
            <td>작성자</td>
            <td>
                <input type="text" name="name" style="width:150px;"
                    value="${boardDTO.name}" />
            </td>
        </tr>
        <tr>
            <td>제목</td>
            <td>
                <input type="text" name="title" style="width:90%;"
                    value="${boardDTO.title}" />
            </td>
        </tr>
        <tr>
            <td>내용</td>
            <td>
                <textarea name="content" style="width:90%;
                    height:100px;">${boardDTO.content}</textarea>
            </td>
        </tr>
        <!-- <tr>
            <td>비밀번호</td>
            <td>
                <input type="password" name="pass" style="width:100px;" />
            </td>
        </tr-->
```

```
</td>
</tr> -->
<tr>
  <td colspan="2" align="center">
    <button type="submit">작성 완료</button>
    <button type="reset">RESET</button>
    <button type="button" onclick="location.href='./list.do';">
      목록 바로가기
    </button>
  </td>
</tr>
</table>
</form>
</body>
```

여기까지 작성하세요.

삭제하기

컨트롤러 : com.edu.springboot.MainController.java

```
20 @PostMapping("/delete.do")
21 public String boardDeletePost(HttpServletRequest req) {
22     int result = dao.delete(req.getParameter("idx"));
23     System.out.println("글삭제결과:" + result);
24     return "redirect:list.do";
25 }
26 }
```

인터페이스 : com.edu.springboot.jdbc.IBoardService.java

```
8 @Mapper
9 public interface IBoardService {
10     //Paging처리를 위해 게시물의 갯수를 카운트
11     public int getTotalCount(ParameterDTO parameterDTO);
12     //게시판 목록(페이징 기능 추가)
13     public ArrayList<BoardDTO> listPage(ParameterDTO parameterDTO);
14     //작성하기
15     public int write(@Param("_name") String name,
16                     @Param("_title") String title,
17                     @Param("_content") String content);
18     //내용보기
19     public BoardDTO view(BoardDTO boardDTO);
20     //수정하기
21     public int edit(BoardDTO boardDTO);
22     //삭제하기
23     public int delete(String idx);
24 }
```

매퍼 : resources/mapper/BoardDAO.xml

```
69 <delete id="delete">
70     DELETE FROM myboard WHERE idx=#{param1}
71 </delete>
72
73 </mapper>
```

여기까지 작성하세요.

퀴즈]

기능1 : 상세보기에 게시물의 조회수를 증가하는 기능을 추가하시오.

기능2 : 목록에서 페이징을 위한 설정값을 하드코딩 해놓은 부분을 프로퍼티 파일에서 읽어올 수 있도록 구현하시오. 단 application.properties 가 아닌 개발자가 직접 정의한 프로퍼티 파일을 사용하시오.