

게시판 구현하기 (템플릿)

게시판 구현하기(템플릿)

• 설명

아래 템플릿 복사 후 파일 생성 (boardDetail.jsp)

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

<!doctype html>
<html lang="ko" class="h-100">
  <head>
    <%@ include file="../common/head.jsp" %>
    <%@ include file="../common/smartereditor.jsp" %>
  </head>

  <body class="d-flex flex-column h-100">
    <!-- Fixed navbar -->
    <%@ include file="../common/header.jsp" %>

    <!-- Begin page content -->
    <main class="flex-shrink-0">
      <div class="container">
        <h3>상세보기</h3>
        <hr>
        <br/>

        </main>
        <%@ include file="../common/footer.jsp" %>
      </body>
    </html>
```

게시판 구현하기 (상세보기-1)

게시판 구현하기 - 상세보기

- boardList.jsp

게시글을 클릭했을 때 상세보기 페이지로 이동하기 위한 링크 설정
DB에서 해당하는 게시글을 조회하기 위해 detail.do로 게시글의 idx 번호를 넘겨줌

```
</c:when>
<c:otherwise>
  <c:forEach var="item" items="${list}">
    <tr onclick="location.href='detail.do?boardIdx=${item.idx}'">
      <td>${row}</td>
      <td>${item.title}</td>
      <td>${item.writer}</td>
      <td>${item.indate}</td>
      <td>${item.count}</td>
    </tr>
    <c:set var="row" value="${row - 1}" />
  </c:forEach>
</c:otherwise>
</c:choose>
```

게시판 구현하기 - 상세보기

- **kr.co.greenart.board.controller.BoardController**

detail.do 요청을 받고 받은 게시글 번호로 게시글 정보(제목, 내용 등) 가져오기

```
@GetMapping("detail.do")
public String detailBoard(@RequestParam(value="boardIdx") int idx, Model model) {
    Board result = boardService.detailBoard(idx);

    if(!Objects.isNull(result)) {
        model.addAttribute("detail", result);
        return "board/boardDetail";
    } else {
        return "common/errorPage";
    }
}
```

게시판 구현하기 - 상세보기

- **kr.co.greenart.board.model.service.BoardService**

detailBoard 사용을 위한 추상 메소드 사용 및 BoardServiceImpl에 오버라이딩

```
// 게시글 상세보기  
Board detailBoard(int idx);
```

게시판 구현하기 - 상세보기

- **kr.co.greenart.board.model.dao.BoardDao**

DB 조회를 위한 설정

```
public Board detailBoard(SqlSessionTemplate sqlSession, int idx) {  
    return sqlSession.selectOne("boardMapper.detailBoard", idx);  
}
```

게시판 구현하기 - 상세보기

- resources/mappers/board/board-mapper.xml

전달받은 idx를 사용하여 게시글 조회

```
<select id="detailBoard" resultMap="boardResultSet">
  SELECT * FROM board
  WHERE B_IDX = #{idx}
</select>
```


게시판 구현하기 - 상세보기

- 설명

스마트 에디터 비활성화 하기 위해 아래 부분 주석 처리 또는 삭제

```
<!doctype html>
<html lang="ko" class="h-100">
  <head>
    <%@ include file="../../../common/head.jsp" %>
    <!-- <%@ include file="../../../common/smartereditor.jsp" %> --%>
  </head>

  <body class="d-flex flex-column h-100">
```

게시판 구현하기 - 상세보기

• 설명

제목, 내용을 가져오기 위한 설정 및 이후 수정하기 위한 action 설정 및 사용자가 수정할 수 없도록 제목과 내용에 readonly 속성 추가

```
<form action="/board/update.do" method="post">
  <div class="mb-3 justify-content-center">
    <input type="text" class="form-control" id="exampleFormControlInput1"
      name="title" value="${detail.title}" placeholder="제목을 입력하세요." readonly>
    </div>
    <div id="marteditor mb-3 justify-content-center">
      <textarea name="content" id="editorTxt"
        class="form-control"
        rows="13" cols="10"
        placeholder="내용을 입력해주세요"
        readonly>${detail.content}</textarea>
    </div>
    <div class="row">
      <div class="col text-center">
        <button type="button" class="btn btn-danger" onclick="history.back()">취소</button>
        <button type="submit" onclick="save();" class="btn btn-primary">수정</button>
      </div>
    </div>
  </form>
```

게시판 구현하기 (상세보기-1)

게시판 구현하기 - 상세보기

- `kr.co.greenart.board.controller.BoardController`

게시글을 클릭했을 때 조회수가 1 올라가도록 설정하는 작업이 필요함

```
@GetMapping("detail.do")
public String detailBoard(@RequestParam(value="boardIdx") int idx, Model mo)
{
    Board result = boardService.detailBoard(idx);
    result.setIdx(idx);

    if(!Objects.isNull(result)) {
        result.setCount(result.getCount() + 1);
        boardService.countBoard(result);

        model.addAttribute("detail", result);
        model.addAttribute("user", session.getAttribute("memberName"));
        return "board/boardDetail";
    }
}
```

가져온 값에 1을 추가하여 DTO에 저장
저장한 값을 DB에 추가하기 위한 메서드 호출

게시판 구현하기 - 상세보기

- **kr.co.greenart.board.model.service.BoardServiceImpl**

인터페이스 및 클래스에 countBoard 메서드 생성

```
public int countBoard(Board bo) {  
    return boardDao.countBoard(sqlSession, bo);  
}
```

게시판 구현하기 - 상세보기

- board-mapper.xml

idx를 조회하여 count 값 증가

```
<update id="countBoard" parameterType="Board">  
    UPDATE board SET b_count = #{count}  
    WHERE b_idx = #{idx}  
</update>
```

게시판 구현하기 (상세보기-2)

게시판 구현하기 - 상세보기

- 설명

글 작성자만 수정 버튼이 활성화 되고, 다른 사용자들은 수정 버튼을 숨기는 작업이 필요함

게시판 구현하기 - 상세보기

- **kr.co.greenart.board.controller.BoardController**

현재 접속한 사용자가 글 작성자인지 확인하기 위해 session에서 이름을 가져옴

```
model.addAttribute("detail", result);  
model.addAttribute("user", session.getAttribute("memberName"));  
return "board/boardDetail";
```

게시판 구현하기 - 상세보기

- boardDetail.jsp

글 작성자만 smarteditor.jsp를 호출

```
<head>
  <%@ include file="../common/head.jsp" %>
  <c:if test="${detail.writer == user}">
    <%@ include file="../common/smarteditor.jsp" %>
  </c:if>
</head>
```

게시판 구현하기 - 상세보기

- boardDetail.jsp

글 작성자가 아닐 경우에만 readonly를 적용하기 위한 삼항연산자 사용

```
<div class="mb-3 justify-content-center">
  <input type="text" class="form-control" id="exampleFormControlInput1"
    name="title" value="${detail.title}" placeholder="제목을 입력하세요."
    ${detail.writer != user ? 'readonly' : ''}>
</div>
```

게시판 구현하기 - 상세보기

- boardDetail.jsp

글 작성자일 경우 textarea로 수정이 가능하도록 하고, 글 작성자가 아닐 경우
div 속성 및 readonly 적용

* div 속성 사용 이유 : HTML 태그가 문자열로 출력되는것을 방지하기 위해

```
<div id="main-editor" id="justfy-content-center">
  <c:choose>
    <c:when test="${detail.writer == user}">
      <textarea name="content" id="editorTxt"
        class="form-control" rows="13" cols="10"
        placeholder="내용을 입력해주세요">
        ${detail.content}
      </textarea>
    </c:when>
    <c:otherwise>
      <div name="content" id="editorTxt"
        class="form-control" rows="13" cols="10"
        placeholder="내용을 입력해주세요" style="height: 300px;"
        readonly>
        ${detail.content}
      </div>
    </c:otherwise>
  </c:choose>
</div>
```

게시판 구현하기 - 상세보기

- **c:if vs c:choose**

c:if → 단순한 조건식 하나를 처리할 때 사용 됨

c:choose → 여러 조건식을 처리할 때 사용 됨 (자바에서의 if-else if-else와 동일)

게시판 구현하기 - 상세보기

- boardDetail.jsp

글 작성자만 수정과 삭제 버튼이 보임

```
<div class="row">
  <div class="col text-center">
    <button type="button" class="btn btn-light" onclick="/member/list.do">취소</button>
    <c:if test="${detail.writer == user}">
      <button type="submit" onclick="save();" class="btn btn-primary">수정</button>
      <button type="submit" onclick="save();" class="btn btn-danger">삭제</button>
    </c:if>
  </div>
</div>
```