15장. 페이지 처리 및 검색

pagination

페이지 처리

게시글 목록

번호	제목	작성자	등록일	조회수
34	행복	admin	2022-09-10 11:09:31	50
33	사랑	admin	2022-09-10 11:09:20	7
32	ruby	admin	2022-09-10 11:08:52	0
31	python	admin	2022-09-10 11:08:40	1
30	c	admin	2022-09-10 11:08:24	7
29	java	admin	2022-09-10 11:08:14	1
28	스프링 부트	cloud	2022-09-10 05:52:06	22
27	긴급 내용	admin	2022-09-10 05:50:44	3
26	공지사항	admin	2022-09-10 05:49:53	3
25	개천절	cloud	2022-09-09 12:01:34	2

1 2 3

글쓰기

페이지 처리

- tbl_board 테이블 수정

```
-- tbl_board 테이블 생성
CREATE TABLE tbl_board(
    bno  NUMBER(5),
    title VARCHAR2(200) NOT NULL,
    writer VARCHAR2(20) NOT NULL,
    content VARCHAR2(2000) NOT NULL,
    regdate DATE DEFAULT SYSDATE,
    updatedate DATE DEFAULT SYSDATE,
    cnt NUMBER(5) DEFAULT 0
);
-- 기본키 제약 조건 이름 설정
ALTER TABLE tbl_board ADD CONSTRAINT pk_board
PRIMARY KEY(bno);
```

페이지 처리

- 식별키(pk)와 인덱스

pk를 생성하면 자동으로 인덱스가 생성됨.

인덱스는 색인(찾아보기)이라는 의미로 이미 정렬된 구조이므로 데이터를 찾아내서 SORT 하는 과정을 생략할 수 있음

(예를 들면 bno가 pk이면 가장 먼저 찾은 데이터는 bno 값이 가장 크거나 작은 값임)

```
SELECT
  /*+ INDEX_DESC(tbl_board pk_board) */
   *
FROM tbl_board;
```

힌트는 SELECT문을 어떻게 처리하는지에 대한 설명글 즉 주석이므로 parsing이 되지 않음

방법 : /* + */ 구문안에 사용

페이지 처리

- 인덱스(INDEX) 사용하기

```
-- 정렬하기(ROWNUM-오름차순, bno-내림차순)
-- bno는 게시글이 삭제된 경우 번호가 없어짐
-- ROWNUM은 실제 데이터는 아니지만, 실제의 데이터를 카운트하여 출력(VIEW)할 수 있음
SELECT ROWNUM, bno, title FROM tbl_board
ORDER BY bno DESC;

-- 힌트(hint) 주기 - 인덱스 사용하여 정렬하기 : 기본키 사용
SELECT /*+ INDEX_DESC(tbl_board pk_board) */ ROWNUM, bno, title
FROM tbl_board;

-- 1페이지 가져오기
SELECT /*+ INDEX_DESC(tbl_board pk_board) */ ROWNUM bno, title
FROM tbl_board
WHERE ROWNUM <= 10;

-- 2페이지 가져올수 없음(0부터 시작해야 하므로)
SELECT /*+ INDEX_DESC(tbl_board pk_board) */ ROWNUM bno, title
FROM tbl_board
WHERE ROWNUM >10 AND ROWNUM <=20;
```

페이지 처리

- 인덱스(INDEX) 사용하기

```
-- 서브쿼리(인라인 뷰) 사용:  11 ~ 20 뽑아내기
-- rownum이 키워드 이므로 rn 별칭 사용
SELECT * FROM
(
    SELECT /*+ INDEX_DESC(tbl_board pk_board) */
        ROWNUM rn, bno, title, writer, content
    FROM tbl_board
    WHERE ROWNUM <= 40
)
WHERE rn > 0;

-- 페이지 처리 SQL
SELECT * FROM
(
    SELECT /*+ INDEX_DESC(tbl_board pk_board) */
        rownum rn, bno, title, writer, content, regdate, updatedate, cnt
    FROM tbl_board
    WHERE ROWNUM <= (2 * 10)
)
WHERE rn > (1-1) * 10;
```

페이지 처리

- 재귀 복사

```
-- 더미 데이터 입력
INSERT INTO tbl_board (bno, title, writer, content)
VALUES (seq.NEXTVAL, '테스트 제목', 'user00', '테스트 내용입니다.');

SELECT * FROM tbl_board ORDER BY bno DESC;

-- 재귀 복사
INSERT INTO tbl_board (bno, title, content, writer)
SELECT seq.nextval, title, content, writer from tbl_board;
```

페이지 처리

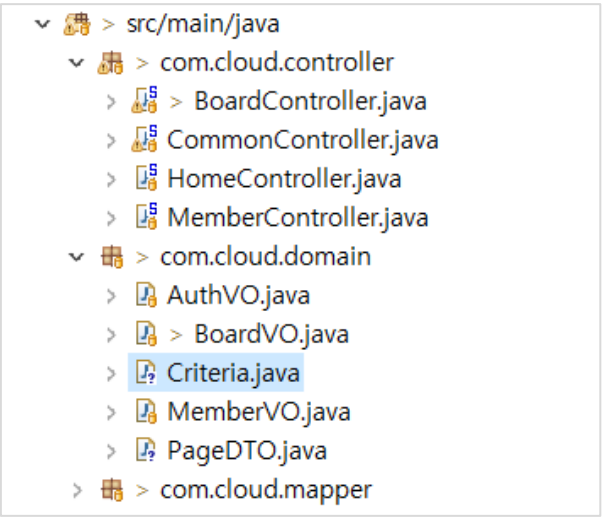
- BoardVO 수정

```
@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 Date updateDate;
    private int cnt;

    private MultipartFile uploadFile;  //파일 업로드
}
```

페이지 처리

- Criteria 클래스

```java
import lombok.Data;

@Data
public class Criteria {
    private int pageNum;    //페이지 번호
    private int amount;     //페이지당 게시글 수

    public Criteria() {     //또 다른 생성자 호출
        this(1, 10);
    }

    public Criteria(int pageNum, int amount) {
        this.pageNum = pageNum;
        this.amount = amount;
    }
}
```



페이지 처리

- BoardMapper 인터페이스

```
public interface BoardMapper {

    public void insert(BoardVO vo);    //글 쓰기

    public List<BoardVO> getBoardList(); //글 목록

    public List<BoardVO> getListWithPage(Criteria cri); //목록 페이지 처리

    public int getTotalCount(Criteria cri);  //게시글 총 개수

    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
}
```

페이지 처리

- BoardMapper.xml

```
<!-- 목록 보기(페이지 처리) -->
<select id="getListWithPage" resultType="com.cloud.domain.BoardVO">
    <![CDATA[
    SELECT * FROM
    (
        SELECT /*+ INDEX_DESC(tbl_board pk_board) */
            ROWNUM rn, bno, title, content, writer, regdate, updatedate, cnt
        FROM tbl_board
        WHERE ROWNUM <= (#{pageNum} * #{amount})
    )
    WHERE rn > (#{pageNum} - 1) * #{amount}
    ]]>
</select>

<!-- 게시글 총 개수 -->
<select id="getTotalCount" resultType="int">
    SELECT COUNT(*) FROM tbl_board WHERE bno > 0
</select>
```

페이지 처리

- CDATA란?

CDATA는 character data(문자 데이터)를 의미하며, 마크업 언어 SGML 및 XML에 사용된다.
이미 정의된 &lt(<), &gt(>) 등의 문자는 마크업에서 해석되기도 어렵다.
이 때, CDATA 영역을 사용하면 어떠한 마크업을 포함하지 않고, 일반 문자열로 인식되도록 할 수 있다.

○ 사용법

```
<![CDATA[
        SQL 문장
 ]]>
```

페이지 처리

- 페이징 테스트 (com.spring.persistence.BoardMapperTests.java)

```java
@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration("file:src/main/webapp/WEB-INF/spring/root-context.xml")
@Log4j
public class BoardMapperTests {

    @Autowired
    private BoardMapper mapper;

    @Test
    public void testGetList() {

        List<BoardVO> list = mapper.getBoardList();

        list.stream()       //스트림 메서드로 리스트 추출함
            .map(board -> board.getBno())  //람다식 사용 - getTitle()로도 테스트
            .forEach(board -> log.info(board));
    }
```

페이지 처리

- 페이징 테스트 - JUnit

bno	title	writer	content	regdate	updatedate	cnt
34	행복	admin	행복	2022-09-10 11:09:31.0	2022-09-10 11:09:31.0	53
33	사랑	admin	사랑	2022-09-10 11:09:20.0	2022-09-10 11:09:20.0	9
31	python	admin	python	2022-09-10 11:08:40.0	2022-09-10 11:08:40.0	4
30	c언어	admin	c언어	2022-09-10 11:08:24.0	2022-09-13 05:32:17.0	10
29	java	admin	java	2022-09-10 11:08:14.0	2022-09-10 11:08:14.0	1
28	스프링 부트	cloud	스프링 부트 시작하기	2022-09-10 05:52:06.0	2022-09-12 11:55:10.0	22
27	긴급 내용	admin	서버 점검	2022-09-10 05:50:44.0	2022-09-10 05:50:44.0	3
26	공지사항	admin	서버 점검	2022-09-10 05:49:53.0	2022-09-10 05:49:53.0	3
23	추석	cloud	추석	2022-09-09 11:58:47.0	2022-09-09 11:58:47.0	23
22	포도	cloud	포도	2022-09-09 05:27:54.0	2022-09-09 05:27:54.0	1
21	바나나	cloud	바나나	2022-09-09 05:27:46.0	2022-09-09 05:27:46.0	0
18	사과	cloud	사과	2022-09-09 05:27:22.0	2022-09-09 05:27:22.0	5
17	고추장	cloud	고추장, 된장	2022-09-09 05:26:30.0	2022-09-09 05:26:30.0	16

페이지 처리

- 페이징 테스트 - JUnit

```
@Test
public void testPaging() {

    Criteria cri = new Criteria();
    //2페이지 10개 출력
    cri.setPageNum(2);    //2페이지
    cri.setAmount(10);    //게시글 수 - 10개

    List<BoardVO> list = mapper.getListWithPage(cri);

    list.stream()
        .map(board -> board.getBno())
        .forEach(board -> log.info(board));
}
```

페이지 처리

- 페이징 테스트 - JUnit

```
|---------|----|--------|---------|-------|----------------------|----------------------|----|
|rn       |bno |title   |content  |writer |regdate               |updatedate            |cnt |
|---------|----|--------|---------|-------|----------------------|----------------------|----|
|[unread] |21  |바나나     |바나나      |cloud  |2022-09-09 05:27:46.0 |2022-09-09 05:27:46.0 |0
|[unread] |18  |사과       |사과        |cloud  |2022-09-09 05:27:22.0 |2022-09-09 05:27:22.0 |5
|[unread] |17  |고추장      |고추장, 된장  |cloud  |2022-09-09 05:26:30.0 |2022-09-09 05:26:30.0 |16
|[unread] |13  |양파       |양파        |cloud  |2022-09-09 05:25:50.0 |2022-09-09 05:25:50.0 |5
|[unread] |12  |마늘       |마늘
슈퍼푸드 |cloud  |2022-09-09 05:25:42.0 |2022-09-09 11:22:27.0 |31  |
|[unread] |10  |토        |토         |cloud  |2022-09-09 05:25:25.0 |2022-09-09 05:25:25.0 |10  |
|[unread] |9   |금  - 불금 |금         |cloud  |2022-09-09 05:25:19.0 |2022-09-13 14:53:06.0 |3
|[unread] |7   |수        |수         |cloud  |2022-09-09 05:25:08.0 |2022-09-09 05:25:08.0 |1   |
|[unread] |6   |화- 화요일  |화         |cloud  |2022-09-09 05:25:01.0 |2022-09-13 15:03:40.0 |1
|[unread] |4   |봄        |봄         |cloud  |2022-09-09 05:24:47.0 |2022-09-09 05:24:47.0 |2   |
|---------|----|--------|---------|-------|----------------------|----------------------|----|
```

페이지 처리

- BoardService 수정

```
public interface BoardService {

    public void insert(BoardVO vo);    //글 쓰기

    //public List<BoardVO> getBoardList(); //글 목록

    public List<BoardVO> getListWithPage(Criteria cri); //목록 페이지 처리

    public int getTotalCount(Criteria cri);  //게시글 총 개수
```

```
@Override
public List<BoardVO> getListWithPage(Criteria cri) { //목록 보기(페이징)
    return mapper.getListWithPage(cri);
}

@Override
public int getTotalCount(Criteria cri) { //총 게시글 수
    return mapper.getTotalCount(cri);
}
```

페이지 처리

- BoardController 수정

```
//게시글 목록(페이징) 요청
@GetMapping("/boardList")
public String getBoardList(Criteria cri, Model model) {
    List<BoardVO> boardList = service.getListWithPage(cri);

    log.info("boardList: " + cri);

    model.addAttribute("boardList", boardList);
    return "/board/boardList";
}
```

페이지 처리

- 페이징 처리를 위한 클래스 설계

  페이지의 끝 번호 계산

  마지막페이지 = (int)(Math.ceil(페이지번호 / 10.0)) * 10

  1 2 3 4 5 6 7 8 9 10

  11 12 13 14 15 16 17 18 19 20

  - 1페이지의 경우: Math.ceil(0.1)*10 = 10
  - 10페이지의 경우 : Math.ceil(1)*10 = 10
  - 11페이지의 경우: Math.ceil(1.1)*10 = 20

  시작페이지 = 마지막페이지 - 9

페이지 처리

- PageDTO 클래스

```
@Data
public class PageDTO {
    private int startPage;    //시작 페이지
    private int endPage;      //마지막 페이지
    private boolean prev, next;  //이전, 다음

    private int total;      //게시글 총 수
    private Criteria cri;  //페이지 번호와 페이지당 게시글 수 참조

    public PageDTO(Criteria cri, int total) {
        this.cri = cri;
        this.total = total;
        //마지막 페이지                          //3/10.0.. 0.3 -> 1(올림) -> 10page
        this.endPage = (int)(Math.ceil(cri.getPageNum() / 10.0)) * 10;
        //시작 페이지
        this.startPage = this.endPage - 9;
        //실제 마지막 페이지                       //74.0/10 -> 7.4 -> 8
        int realEndPage = (int)(Math.ceil((total * 1.0) / cri.getAmount()));

        //마지막 페이지와 실제 페이지 조건
        this.endPage = realEndPage < endPage ? realEndPage : endPage;

        this.prev = this.startPage > 1;    //다음
        this.next = this.endPage < realEndPage;  //이전
    }
}
```

페이지 처리

- 게시글 총 개수 구하기

BoardMapper.java

```java
public int getTotalCount(Criteria cri);  //게시글 총 개수
```

BoardMapper.xml

```xml
<!-- 게시글 총 개수 -->
<select id="getTotalCount" resultType="int">
    SELECT COUNT(*) FROM tbl_board
</select>
```

BoardService.java

```java
public int getTotalCount(Criteria cri);  //게시글 총 개수
```

BoardServiceImpl.java

```java
@Override
public int getTotalCount(Criteria cri) { //총 게시글 수
    return mapper.getTotalCount(cri);
}
```

페이지 처리

- 페이징 테스트

```
@Test
public void testPageDTO() {
    Criteria cri = new Criteria();
    cri.setPageNum(1);
    //cri.setPageNum(11);

    PageDTO page = new PageDTO(cri, 61);  //total=60
    log.info(page);
}
```

```
 PageDTO(startPage=1, endPage=7, prev=false, next=false, total=61, cri=Criteria(pageNum=1, amount=10))
riPool-1 - Shutdown initiated...
riPool-1 - Shutdown completed.
```

페이지 처리

- BoardController 수정

```
//게시글 목록(페이지 요청)
@GetMapping("/boardList")
public String getBoardList(Criteria cri, Model model) {
    List<BoardVO> boardList = service.getListWithPage(cri);
    PageDTO pageMaker = new PageDTO(cri, service.getTotalCount(cri));

    log.info(cri);
    model.addAttribute("boardList", boardList); //model-"boardList"
    model.addAttribute("pageMaker", pageMaker); //model-"pageMaker"
    return "/board/boardList";
}
```

페이지 처리

- JSP 목록 페이지에서 번호 출력

boardList.jsp

```
</table>
<!-- 페이지 번호 -->
<div>
    <ul>
        <c:if test="${pageMaker.prev}">
        <li class="page-link"><a href="${pageMaker.startPage - 1}">이전</a></li>
        </c:if>
        <c:forEach begin="${pageMaker.startPage}"
                   end="${pageMaker.endPage}" var="num">
            <!-- 현재 페이지 활성화 -->
            <c:if test="${pageMaker.cri.pageNum eq num}">
                <li class="page-link"><a href="${num}">
                    <b><c:out value="${num}" /></b></a>
                </li>
            </c:if>
            <c:if test="${pageMaker.cri.pageNum ne num}">
                <li class="page-link">
                    <a href="${num}"><c:out value="${num}" /></a>
                </li>
            </c:if>
        </c:forEach>
        <c:if test="${pageMaker.next}">
        <li class="page-link"><a href="${pageMaker.endPage + 1}">다음</a></li>
        </c:if>
    </ul>
</div>
```

페이지 처리

- 페이지 처리 폼 만들기

boardList.jsp

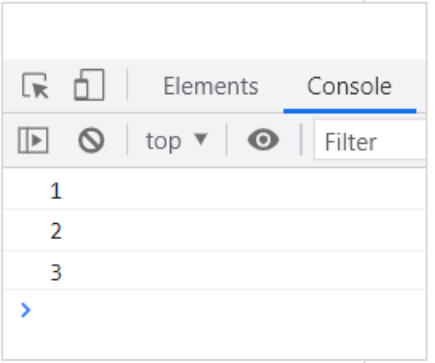
```
<!-- 페이지 처리 전송 폼(get 방식) -->
<form action="/board/boardList" method="get" id="actionForm">
    <input type="hidden" name="pageNum" value="${pageMaker.cri.pageNum}">
    <input type="hidden" name="amount" value="${pageMaker.cri.amount}">
</form>

<!-- 글쓰기 버튼 -->
```

페이지 처리

- 이벤트처리 - 제이쿼리로 폼 전송

boardList.jsp

```
<script type="text/javascript">
    //페이징 처리 - 목록 페이지
    //jQuery 라이브러리 임포트함
    $(document).ready(function(){

        let actionForm = $("#actionForm"); //폼 선택

        $(".page-link a").on('click', function(e){

            e.preventDefault();  //기본 동작 제한(링크 안되게 함)
            let targetPage = $(this).attr("href");
            console.log(targetPage);

            actionForm.find("input[name='pageNum']").val(targetPage);
            //actionForm.submit(); //페이지 처리 완료(submit 전에 콘솔에서 테스트 하기)
        });
```



페이지 처리

제목 Search

번호	제목	작성자	등록일	조회수
38	가입 인사	cloud	2022-09-16 10:18:18	0
37	테스트 제목	user00	2022-09-16 10:18:18	0
36	테스트 제목	user00	2022-09-16 10:18:18	0
35	가입 인사	cloud	2022-09-16 10:18:18	0
34	테스트 제목	user00	2022-09-16 10:18:18	0
33	테스트 제목	user00	2022-09-16 10:18:18	0
32	가입 인사	cloud	2022-09-16 10:18:13	0
31	가입 인사	cloud	2022-09-16 10:18:13	0
30	가입 인사	cloud	2022-09-16 10:18:13	0
29	가입 인사	cloud	2022-09-16 10:18:13	0

1 2 3 4 5 6 7 8 9 **10** 다음

글쓰기

페이지 처리

제목 Search

번호	제목	작성자	등록일	조회수
28	테스트 제목	user00	2022-09-16 10:18:13	0
27	테스트 제목	user00	2022-09-16 10:18:13	0
26	가입 인사	cloud	2022-09-16 10:18:13	0
25	테스트 제목	user00	2022-09-16 10:18:13	0
24	테스트 제목	user00	2022-09-16 10:18:13	0
23	가입 인사	cloud	2022-09-16 10:18:13	0
22	가입 인사	cloud	2022-09-16 10:18:13	0
21	테스트 제목	user00	2022-09-16 10:18:13	0
20	테스트 제목	user00	2022-09-16 10:18:13	0
19	가입 인사	cloud	2022-09-16 10:18:13	0

이전 **11** 12 13

글쓰기

페이지 처리

- 상세 페이지로 이동 – 이벤트 처리

boardList.jsp

```
<tr>
    <td><c:out value="${board.bno}"/></td>
    <td>
        <a class="move" href='<c:out value="${board.bno}"/>'>
            <c:out value="${board.title}" /></a>
    </td>
```

```
//상세 페이지
$(".move").on("click", function(e){
    e.preventDefault();  //기본 동작 제한
    let targetBno = $(this).attr("href");
    console.log(targetBno);

    actionForm.append("<input type='hidden' name='bno' value='" + targetBno + "'>");
    actionForm.attr("action", "/board/boardView");
    actionForm.submit();
});
}); //$(document) 끝
```

페이지 처리

- 상세 보기 페이지 처리하기

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

    model.addAttribute("board", board); //model-"board"
    model.addAttribute("cri", cri);  //model-"cri"
    return "/board/boardView";
}
```

페이지 처리

- 상세 보기 -> 목록 페이지 이동시 번호 유지

  상세 보기 화면으로 이동한 후 다시 목록으로 돌아올때 해당 페이지로 가지 않고 무조건 1 페이지로 이동하는 상태가 유지되는 문제가 발생함.

```
                </security:authorize>
                    <a href="/board/boardList"><input type="button" value="목록" class="listBtn">
                </td>
            </tr>
        </table>
    </form>
</section>
<form action="/board/boardList" method="get" id="actionForm">
    <input type="hidden" name="bno" value="${board.bno}">
    <!-- 목록 페이지로 이동시 페이지 번호 유지(없으면 1페이지로 감) -->
    <input type="hidden" name="pageNum" value="${cri.pageNum}">
    <input type="hidden" name="amount" value="${cri.amount}">
</form>
```

boardView.jsp

페이지 처리

- 상세 보기 페이지 -> 목록 페이지로 이동

boardView.jsp

```
<script type="text/javascript">

    $(document).ready(function(){

        let actionForm = $("#actionForm");
        //목록 버튼 클릭
        $(".listBtn").click(function(e){

            e.preventDefault();
            actionForm.submit();
        });
    });
</script>
```

페이지 처리

- 수정/삭제 시에 페이지 처리

```
@PostMapping("/board/updateBoard")
public String updateBoard(BoardVO vo, Criteria cri,
        RedirectAttributes rttr) {  //글 수정 요청
    service.update(vo);
    //URL 뒤에 원래의 페이지로 이동하기 위해 pageNum과 amount 값을 보내줌
    rttr.addAttribute("pageNum", cri.getPageNum());
    rttr.addAttribute("amount", cri.getAmount());

    return "redirect:/board/boardList";
}
```

RedirectAttributes는 리다이렉트가 발생하기 전에 모든 플래시 속성을 세션에 복사한다. 리다이렉션 이후에는 저장된 플래시 속성을 세션에서 모델로 이동시킨다.

페이지 처리

```
public interface RedirectAttributes extends Model {

    @Override
    RedirectAttributes addAttribute(String attributeName, @Nullable

    @Override
    RedirectAttributes addAttribute(Object attributeValue);

    @Override
    RedirectAttributes addAllAttributes(Collection<?> attributeValu
```

```
@GetMapping("/board/deleteBoard")
public String deleteBoard(BoardVO vo, Criteria cri,
        RedirectAttributes rttr) {  //글 삭제 요청
    service.delete(vo);
    rttr.addAttribute("pageNum", cri.getPageNum());
    rttr.addAttribute("amount", cri.getAmount());

    return "redirect:/board/boardList";
}
```

페이지 처리

- 수정/삭제 시에 페이지 처리

boardView.jsp

```
<div class="title">
    <h2>글 상세보기</h2>
</div>
<form action="/board/updateBoard" method="post">
    <!-- 수정 시엔 기본키인 bno를 반드시 명시해 줌 -->
    <input type="hidden" name="bno" value="${board.bno}">
    <!-- 수정, 삭제시 페이지 번호 유지(없으면 1페이지 이동) -->
    <input type="hidden" name="pageNum" value="${cri.pageNum}">
    <input type="hidden" name="amount" value="${cri.amount}">
    <table class="tbl_view">
        <tr>
```

검색 처리

게시글 목록

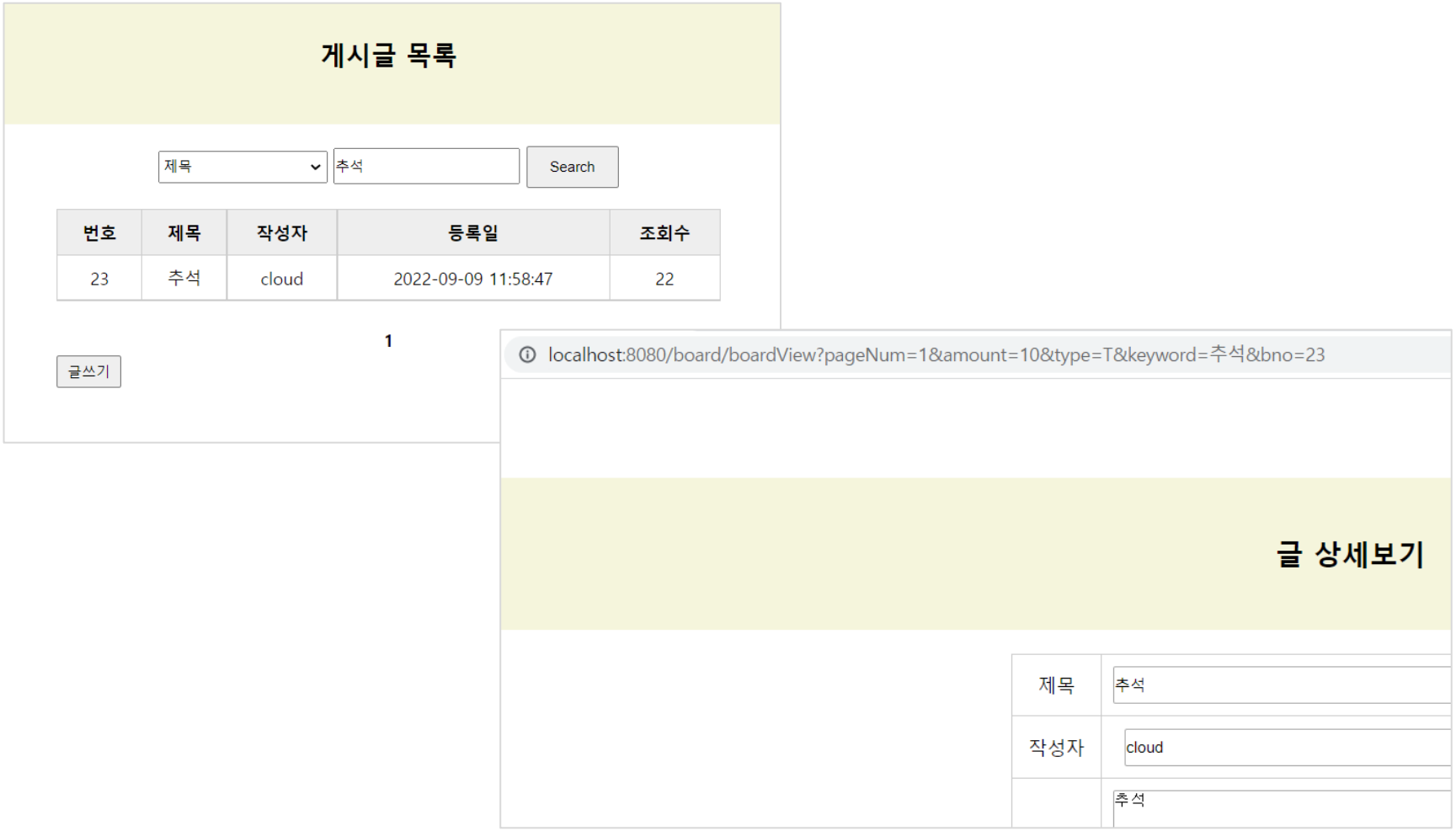
제목 추석 Search

번호	제목	작성자	등록일	조회수
34	행복	admin	2022-09-10 11:09:31	53
33	사랑	admin	2022-09-10 11:09:20	8
31	python	admin	2022-09-10 11:08:40	4
30	c언어	admin	2022-09-10 11:08:24	10
29	java	admin	2022-09-10 11:08:14	1
28	스프링 부트	cloud	2022-09-10 05:52:06	22
27	긴급 내용	admin	2022-09-10 05:50:44	3
26	공지사항	admin	2022-09-10 05:49:53	3
23	추석	cloud	2022-09-09 11:58:47	22
22	포도	cloud	2022-09-09 05:27:54	1

1 2 3

글쓰기

검색 처리


게시글 목록
제목
추석
Search
번호
제목
작성자
등록일
조회수
23
추석
cloud
2022-09-09 11:58:47
22
1
글쓰기
localhost:8080/board/boardView?pageNum=1&amount=10&type=T&keyword=추석&bno=23
글 상세보기
제목
추석
작성자
cloud
추석

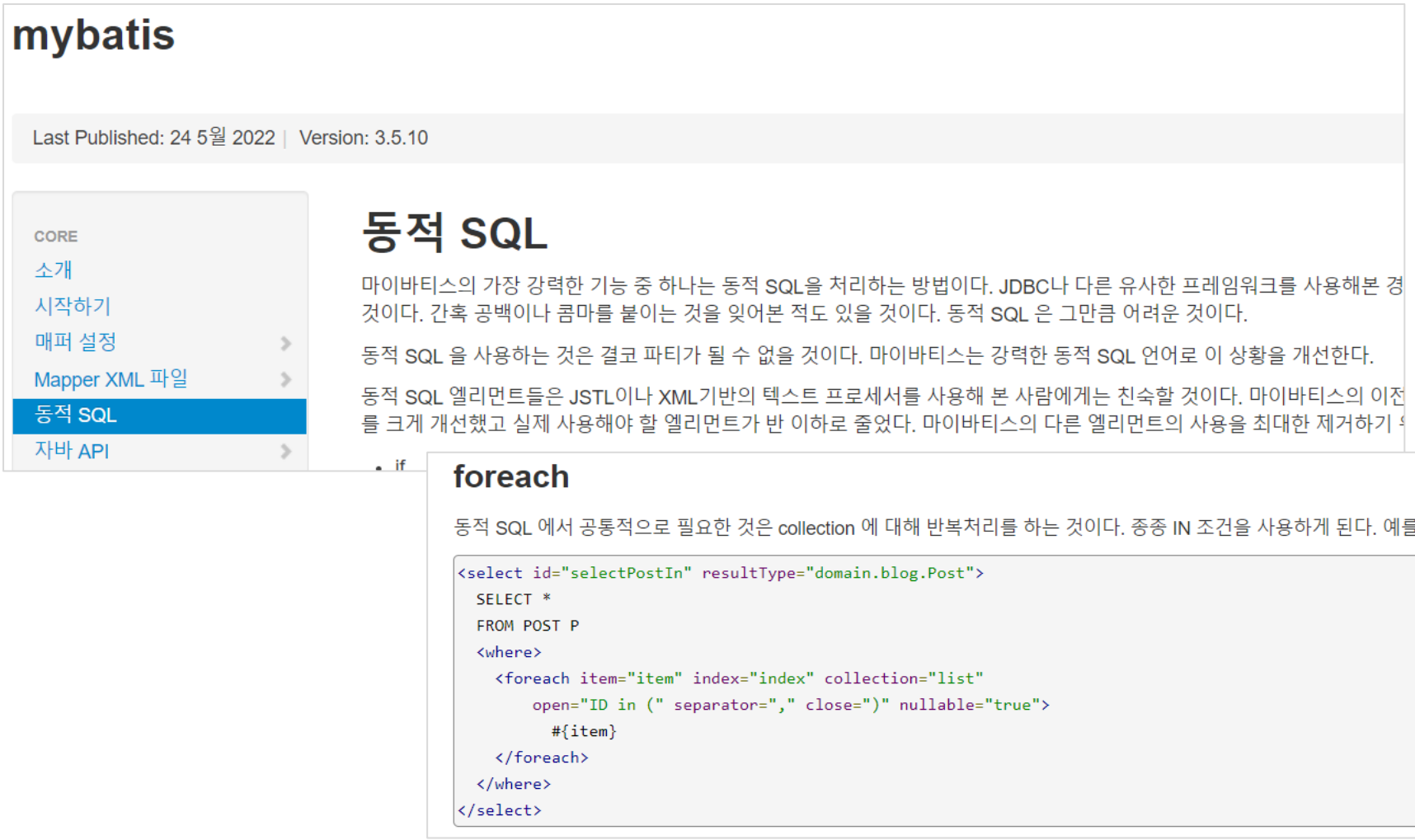
검색 처리

- 검색 처리 테스트 Mapper 설계

```
public interface BoardMapper {

    public void insert(BoardVO vo);    //글 쓰기

    public List<BoardVO> getBoardList(); //글 목록

    public List<BoardVO> getListWithPage(Criteria cri); //목록 페이지 처리

    public int getTotalCount(Criteria cri);  //게시글 총 개수

    List<BoardVO> searchTest(Map<String, Map<String, String>> map);  //검색
```

검색 처리

- MyBatis의 동적 SQL – mybatis document로 검색



검색 처리

- 검색 처리 테스트

BoardMapperTests.java

```
@Test
public void testSearch() {
    Map<String, String> map = new HashMap<>();
    map.put("T", "추석");
    map.put("C", "java");
    map.put("W", "cloud");

    //중첩 Map 사용
    Map<String, Map<String, String>> outer = new HashMap<>();
    outer.put("map", map);  //xml쪽의 collection="map"

    List<BoardVO> list = mapper.searchTest(outer);

    log.info(list);
}
```

검색 처리

- 검색 처리 테스트

BoardMapper.xml

```
<!-- 게시글 검색(테스트용) -->
<select id="searchTest" resultType="com.cloud.domain.BoardVO">
    <![CDATA[
        SELECT * FROM tbl_board
        WHERE
    ]]>
        <foreach collection="map" index="key" item="val">
            <!-- #{key} #{val} 키 값 -->
            <if test="key=='T'.toString()">
                title like #{val}
            </if>
        </foreach>

    <![CDATA[
          ROWNUM < 10
    ]]>
</select>
```

※ 현재 error임

```
ver.T4CTTIoer11.processError(T4CTTIoer11.java:637)

PreparedStatement.execute() FAILED! SELECT * FROM tbl_board WHERE title like '가을' ROWNUM < 10

eption: ORA-00933: SQL 명령어가 올바르게 종료되지 않았습니다
```

검색 처리

- 검색 처리 테스트

```
<!-- 게시글 검색(테스트용) -->
<select id="searchTest" resultType="com.cloud.domain.BoardVO">
    <![CDATA[
        SELECT * FROM tbl_board
        WHERE
    ]]>
        <trim suffix="AND">
        <foreach collection="map" index="key" item="val">
            <!-- #{key} #{val} 키 값 -->
            <if test="key=='T'.toString()">
                title like #{val}
            </if>
        </foreach>
        </trim>
    <![CDATA[
          ROWNUM < 10
    ]]>
</select>
```

※ error 해결

```
jdbc.audit - 1. PreparedStatement.setString(1, "가을") returned
jdbc.sqlonly - SELECT * FROM tbl_board WHERE title like '가을' AND ROWNUM < 10
```

검색 처리

- 검색 처리 테스트

```
        <trim suffix="AND">
        <foreach collection="map" index="key" item="val" >
            <!-- #{key} #{val} 키 값 -->
            <if test="key=='T'.toString()">
                title like #{val}
            </if>
            <if test="key=='C'.toString()">
                content like #{val}
            </if>
            <if test="key=='W'.toString()">
                writer like #{val}
            </if>
        </foreach>
        </trim>
```

※ 현재 error임

```
SQL 명령어가 올바르게 종료되지 않았습니다

ber11.java:637)

! SELECT * FROM tbl_board WHERE content like '추석' title like '가을' writer like 'cloud' AND ROWNUM
```

검색 처리

- 검색 처리 테스트

```
<trim suffix="AND">
<foreach collection="map" index="key" item="val"
         separator="OR" open="(" close=")">
    <!-- #{key} #{val} 키 값 -->
    <if test="key=='T'.toString()">
        title like #{val}
    </if>
    <if test="key=='C'.toString()">
        content like #{val}
    </if>
    <if test="key=='W'.toString()">
        writer like #{val}
    </if>
</foreach>
</trim>
```

※ error 해결

```
INFO : jdbc.audit - 1. PreparedStatement.setString(1, "추석") returned
INFO : jdbc.audit - 1. PreparedStatement.setString(2, "가을") returned
INFO : jdbc.audit - 1. PreparedStatement.setString(3, "cloud") returned
INFO : jdbc.sqlonly - SELECT * FROM tbl_board WHERE ( content like '추석' OR title like '가을' OR writer like 'cloud'
) AND ROWNUM < 10
```

검색 처리

- 검색 처리 테스트

```
bno |title    |writer |content        |regdate                |updatedate
----|--------|-------|---------------|-----------------------|-----------------
1   |겨울       |cloud  |겨울             |2022-09-09 05:23:53.0 |2022-09-09 05:23
2   |가을       |cloud  |가을             |2022-09-09 05:24:02.0 |2022-09-09 05:24
3   |여름       |cloud  |여름             |2022-09-09 05:24:11.0 |2022-09-13 15:11
4   |봄        |cloud  |봄              |2022-09-09 05:24:47.0 |2022-09-09 05:24:
6   |화- 화요일   |cloud  |화              |2022-09-09 05:25:01.0 |2022-09-13 15:03
7   |수        |cloud  |수              |2022-09-09 05:25:08.0 |2022-09-09 05:25:
9   |금  - 불금 |cloud  |금              |2022-09-09 05:25:19.0 |2022-09-13 14:53
10  |토        |cloud  |올해 추석은
트요일이네 |2022-09-09 05:25:25.0 |2022-09-14 06:15:42.0 |11   |
12  |마늘       |cloud  |마늘
슈퍼푸드       |2022-09-09 05:25:42.0 |2022-09-09 11:22:27.0 |31   |
----|--------|-------|---------------|-----------------------|-----------------
```

검색 처리

- Criteria 클래스 수정

```
@Data
public class Criteria {
    private int pageNum;    //페이지 번호
    private int amount;     //페이지당 게시글 수

    private String type;  //t, tc, tcw (배열, 맵으로 구현)
    private String keyword;  //검색어

    public Criteria() {     //또 다른 생성자 호출
        this(1, 10);
    }

    public Criteria(int pageNum, int amount) {
        this.pageNum = pageNum;
        this.amount = amount;
    }

    //타입을 저장할 메서드 정의(type이 null이 아니면 문자열을 분리하여 배열로 반환)
    public String[] getTypeArr() {
        return type == null ? new String[] {} : type.split("");
    }
}
```

검색 처리

```
<select id="getListWithPage" resultType="com.cloud.domain.BoardVO">
    <![CDATA[
    SELECT * FROM
    (
        SELECT /*+ INDEX_DESC(tbl_board pk_board) */
            ROWNUM rn, bno, title, content, writer, regdate, updatedate, cnt
        FROM tbl_board
        WHERE
    ]]>
        <!-- (title like ...) and -->
        <foreach collection="typeArr" item="type">
            <if test="type='T'.toString()">
                title like #{keyword}
            </if>
            <if test="type='C'.toString()">
                content like #{keyword}
            </if>
            <if test="type='W'.toString()">
                writer like #{keyword}
            </if>
        </foreach>
    <![CDATA[
        ROWNUM <= (#{pageNum} * #{amount})
    )
    WHERE rn > (#{pageNum} - 1) * #{amount}
    ]]>
</select>
```

검색 처리

- 검색 및 페이징 처리 테스트

```java
@Test
public void testSearchPaging() {
    //type과 keyword 입력
    Criteria cri = new Criteria();
    cri.setType("T");
    //cri.setType("TC");
    cri.setKeyword("추석");

    List<BoardVO> list = mapper.getListWithPage(cri);

    list.forEach(board -> Log.info(board));
}
```

※ 현재 error임

```
ERROR: jdbc.audit - 1. PreparedStatement.execute() SELECT * FROM ( SELECT /*+ INDEX_DESC(tbl_board pk_board)
writer, regdate, updatedate, cnt FROM tbl_board WHERE title like '추석' content like '추석' writer
like '추석' ROWNUM <= (1 * 10) ) WHERE rn > (1 - 1) * 10

java.sql.SQLSyntaxErrorException: ORA-00907: 누락된 우괄호
```

검색 처리

- 검색 및 페이징 처리 테스트

```
<!-- (title like ...) and -->
<trim prefix="(" suffix=") AND">
<foreach collection="typeArr" item="type" separator="OR">
    <if test="type=='T'.toString()">
        title like #{keyword}
    </if>
    <if test="type=='C'.toString()">
        content like #{keyword}
    </if>
    <if test="type=='W'.toString()">
        writer like #{keyword}
    </if>
</foreach>
</trim>
```

※ error 해결

```
INFO : jdbc.sqlonly - SELECT * FROM ( SELECT /*+ INDEX_DESC(tbl_board pk_board) */ ROWNUM rn, bno,
writer, regdate, updatedate, cnt FROM tbl_board WHERE ( title like '추석' OR content like '추석'
) AND ROWNUM <= 1 * 10 ) WHERE rn > (1 - 1) * 10
```

검색 처리

- 문자 포함 처리('%' 사용)

```
<!-- (title like ...) and -->
<trim prefix="(" suffix=") AND">
<foreach collection="typeArr" item="type" separator="OR">
    <if test="type=='T'.toString()">
        title like '%'||#{keyword}||'%'
    </if>
    <if test="type=='C'.toString()">
        content like '%'||#{keyword}||'%'
    </if>
    <if test="type=='W'.toString()">
        writer like '%'||#{keyword}||'%'
    </if>
</foreach>
</trim>
```

검색 처리

- 페이징 및 검색 테스트 처리

```java
@Test
public void testSearchPaging() {
    //type과 keyword 입력
    Criteria cri = new Criteria();
    cri.setType("TC");
    cri.setKeyword("추석");

    List<BoardVO> list = mapper.getListWithPage(cri);

    list.forEach(board -> Log.info(board));
}
```

```
INFO : jdbc.resultsettable -
|---------|----|------|--------------|-------|----------------------|------------
|rn       |bno |title |content       |writer |regdate               |updatedate
|---------|----|------|--------------|-------|----------------------|------------
|[unread] |23  |추석    |추석            |cloud  |2022-09-09 11:58:47.0 |2022-09-09
|[unread] |10  |토     |올해 추석은
토요일이네 |cloud  |2022-09-09 05:25:25.0 |2022-09-14 06:15:42.0 |11  |
|---------|----|------|--------------|-------|----------------------|------------
```

검색 처리

- 검색 모듈화 하기

```
<!-- 검색 모듈 -->
<sql id="criteria">
    <trim prefix="(" suffix=") AND">
        <foreach collection="typeArr" item="type" separator="OR">
            <if test="type=='T'.toString()">
                title like '%'||#{keyword}||'%'
            </if>
            <if test="type=='C'.toString()">
                content like '%'||#{keyword}||'%'
            </if>
            <if test="type=='W'.toString()">
                writer like '%'||#{keyword}||'%'
            </if>
        </foreach>
    </trim>
</sql>
```

검색 처리

- 목록 보기 및 총 게시글수 두 곳에 모듈 포함

```
<select id="getListWithPage" resultType="com.cloud.domain.BoardVO">
    <![CDATA[
    SELECT * FROM
    (
       SELECT /*+ INDEX_DESC(tbl_board pk_board) */
           ROWNUM rn, bno, title, content, writer, regdate, updatedate, cnt
        FROM tbl_board
        WHERE
    ]]>
        <include refid="criteria" />
    <![CDATA[
        ROWNUM <= #{pageNum} * #{amount}
    )
    WHERE rn > (#{pageNum} - 1) * #{amount}
    ]]>
</select>
```

검색 처리

- 목록 보기 및 총 게시글수 두 곳에 모듈 포함

```
<!-- 게시글 총 개수(검색) -->
<select id="getTotalCount" resultType="int">
    <![CDATA[
    SELECT COUNT(*) FROM tbl_board
    WHERE
    ]]>
        <include refid="criteria" />
    <![CDATA[
    bno > 0
    ]]>
</select>
```

검색 처리

- BoardController 수정

```
public class BoardController {

    @Autowired
    private BoardService service;

    @GetMapping("/boardList")
    public String getBoardList(Criteria cri, Model model) {//게시글 목록 요청
        List<BoardVO> boardList = service.getListWithPage(cri);
        PageDTO pageMaker = new PageDTO(cri, service.getTotalCount(cri));

        log.info(cri);
        model.addAttribute("boardList", boardList); //model-"boardList"
        model.addAttribute("pageMaker", pageMaker);
        return "/board/boardList";
    }
```

검색 처리

- BoardController 수정

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

인증 - 상세 보기 시 로그인 페이지로 이동

@ModelAttribute()를 사용하여 "cri"를 model로 view로 보내줌

검색 처리

- BoardController 수정

```
@GetMapping("/board/deleteBoard")
public String deleteBoard(BoardVO vo, Criteria cri,
        RedirectAttributes rttr) {  //글 삭제 요청
    service.delete(vo);
    rttr.addAttribute("pageNum", cri.getPageNum());
    rttr.addAttribute("amount", cri.getAmount());
    rttr.addAttribute("type", cri.getType());
    rttr.addAttribute("keyword", cri.getKeyword());

    return "redirect:/board/boardList";
}

@PostMapping("/board/updateBoard")
public String updateBoard(BoardVO vo, Criteria cri,
        RedirectAttributes rttr) {  //글 수정 요청
    service.update(vo);
    rttr.addAttribute("pageNum", cri.getPageNum());
    rttr.addAttribute("amount", cri.getAmount());
    rttr.addAttribute("type", cri.getType());
    rttr.addAttribute("keyword", cri.getKeyword());

    return "redirect:/board/boardList";
}
```

검색 처리

- 목록 페이지에서 검색 조건 처리

boardList.jsp

```
<div class="title">
    <h2>게시글 목록</h2>
</div>
<!-- 검색 폼 -->
<form action="/board/boardList" method="get" id="searchForm">
    <select name="type">
        <option value="T">제목</option>
        <option value="C">내용</option>
        <option value="W">작성자</option>
        <option value="TC">제목 or 내용</option>
        <option value="TW">제목 or 작성자</option>
        <option value="TWC">제목 or 내용 or 작성자</option>
    </select>
    <input type="text" name="keyword" value="${pageMaker.cri.keyword}"
           class="keyword">
    <input type="hidden" name="pageNum" value="${pageMaker.cri.pageNum}">
    <input type="hidden" name="amount" value="${pageMaker.cri.amount}">
    <button>Search</button>
</form>
<table class="tbl_list">
```

검색 처리

- 목록 페이지에서 검색 조건 처리

boardList.jsp

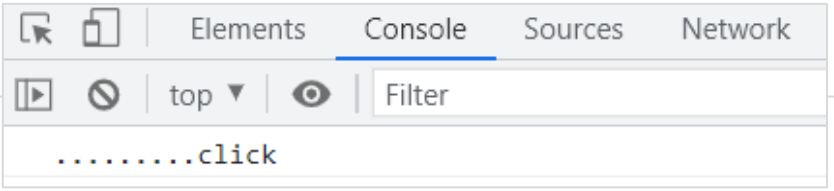
```
<!-- 폼 전송(페이지, 키워드) -->
<form action="/board/boardList" method="get" id="actionForm">
    <input type="hidden" name="pageNum" value="${pageMaker.cri.pageNum}">
    <input type="hidden" name="amount" value="${pageMaker.cri.amount}">
    <input type="hidden" name="type" value="${pageMaker.cri.type}">
    <input type="hidden" name="keyword" value="${pageMaker.cri.keyword}">
</form>
<div class="btn_box">
    <a href="/board/insertBoard"><input type="button" value="글쓰기"></a>
</div>
```

검색 처리

- 목록 페이지에서 검색 조건 처리 이벤트

boardList.jsp

```
<script type="text/javascript">
    //페이징 처리
    $(document).ready(function(){

        /* -- 생략 --  */

        //검색 폼 처리
        let searchForm = $("#searchForm");
        $("#searchForm button").on("click", function(e){
            e.preventDefault();
            console.log("click");

            searchForm.find("input[name='pageNum']").val(1);  //1페이지부터 검색
            searchForm.submit();
        });
    });
</script>
```



검색 처리

- 상세보기 페이지에서 검색 조건 처리

boardView.jsp

```
<div class="title">
    <h2>글 상세보기</h2>
</div>
<form action="/board/updateBoard" method="post">
    <!-- 수정 시엔 기본키인 bno를 반드시 명시해 줌 -->
    <input type="hidden" name="bno" value="${board.bno}">
    <!-- 수정, 삭제시 페이지 번호 유지(없으면 1페이지 이동) 및 검색어 유지-->
    <input type="hidden" name="pageNum" value="${cri.pageNum}">
    <input type="hidden" name="amount" value="${cri.amount}">
    <input type="hidden" name="type" value="${cri.type}">
    <input type="hidden" name="keyword" value="${cri.keyword}">
    <table class="tbl_view">
```