

## 15장. 페이지 처리 및 검색



*pagination*



## 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,
    update date 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

- 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 ROWNUM rn, bno, title, writer, content
  FROM tbl_board
  WHERE ROWNUM <= 10
)
WHERE rn > 0;

-- 페이징 공식 SQL
SELECT * FROM
(
  SELECT
    rownum rn, bno, title, writer, content, regdate, updatedate, cnt
  FROM tbl_board
  WHERE ROWNUM <= (2 * 10)
)
WHERE rn > (1-1) * 10;
```

# 페이지 처리

- Criteria 클래스

```
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;
    }
}
```

```
src/main/java
├── com.cloud.controller
│   ├── BoardController.java
│   ├── CommonController.java
│   ├── HomeController.java
│   └── MemberController.java
├── com.cloud.domain
│   ├── AuthVO.java
│   ├── BoardVO.java
│   ├── Criteria.java
│   ├── MemberVO.java
│   └── PageDTO.java
└── com.cloud.mapper
```



# 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>

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



# 페이지 처리

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

```
INFO : com.zaxxer.hikari.HikariDataSource - HikariPool-1 - Starting...
INFO : com.zaxxer.hikari.HikariDataSource - HikariPool-1 - Start completed.
INFO : com.spring.persistence.BoardMapperTests - 34
INFO : com.spring.persistence.BoardMapperTests - 33
INFO : com.spring.persistence.BoardMapperTests - 31
INFO : com.spring.persistence.BoardMapperTests - 30
INFO : com.spring.persistence.BoardMapperTests - 29
INFO : com.spring.persistence.BoardMapperTests - 28
INFO : com.spring.persistence.BoardMapperTests - 27
INFO : com.spring.persistence.BoardMapperTests - 26
INFO : com.spring.persistence.BoardMapperTests - 23
INFO : com.spring.persistence.BoardMapperTests - 22
INFO : com.spring.persistence.BoardMapperTests - 21
INFO : com.spring.persistence.BoardMapperTests - 18
```



# 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));
}
```

```
INFO : com.zaxxer.hikari.HikariDataSource - HikariPool-1 - Start completed.
INFO : com.spring.persistence.BoardMapperTests - 21
INFO : com.spring.persistence.BoardMapperTests - 18
INFO : com.spring.persistence.BoardMapperTests - 17
INFO : com.spring.persistence.BoardMapperTests - 13
INFO : com.spring.persistence.BoardMapperTests - 12
INFO : com.spring.persistence.BoardMapperTests - 10
INFO : com.spring.persistence.BoardMapperTests - 9
INFO : com.spring.persistence.BoardMapperTests - 7
INFO : com.spring.persistence.BoardMapperTests - 6
```



# 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);  
}
```



- 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** =  $(\text{int})(\text{Math.ceil}(\text{페이지번호} / 10.0)) * 10$

1 2 3 4 5 6 7 8 9 10

11 12 13 14 15 16 17 18 19 20

- 1페이지의 경우:  $\text{Math.ceil}(0.1) * 10 = 10$
- 10페이지의 경우 :  $\text{Math.ceil}(1) * 10 = 10$
- 11페이지의 경우:  $\text{Math.ceil}(1.1) * 10 = 20$

**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

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

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

BoardMapper.xml

BoardService.java

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

BoardServiceImpl.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

```
</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>
```

boardList.jsp



# 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

- 제이쿼리로 폼 전송

boardList.jsp

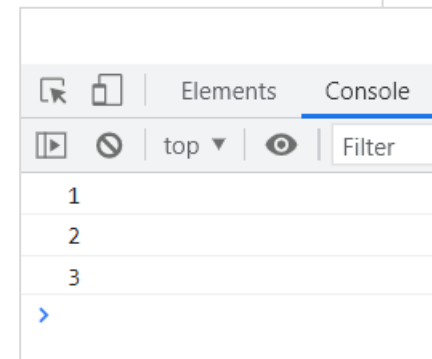
```
<script type="text/javascript">
  //페이징 처리 - 목록 페이지
  //jQuery 라이브러리 임포트함
  $(document).ready(function(){

    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

- 상세 페이지로 이동

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

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

boardView.jsp

```
</c:if>
</security:authorize>
    <a href="/board/boardList"><input type="button" value="목록" class="ListBtn"></a>
</td>
</tr>
</table>
</form>
</section>
<form action="/board/boardList" method="get" id="actionForm">
    <input type="hidden" name="bno" value="${board.bno}">
    <input type="hidden" name="pageNum" value="${cri.pageNum}">
    <input type="hidden" name="amount" value="${cri.amount}">
</form>
```





# 페이지 처리

- 상세 보기 페이지 처리하기

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

    $(document).ready(function(){

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

            e.preventDefault();
            actionForm.submit();

        });
    });
</script>
```

boardView.jsp

# 페이지 처리

- 상세 보기 페이지 처리하기

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="${board.bno}">
    <input type="hidden" name="pageNum" value="${cri.pageNum}">
    <input type="hidden" name="amount" value="${cri.amount}">
    <table class="tbl_view">
        <tr>
            <td width="100">제목</td>
```