Chapter

09

게시판에 페이징 기능 넣기

학습 목표 및 순서

▣ 학습 목표

. 8장에서 만든 게시판에 페이징 기능을 추가합니다. 페이징이란 목록이 길 때 페 이지별로 나눠 보여주는 기능.

▣ 학습 순서

▣ 활용 사례

. 페이징은 게시판뿐 아니라 쇼핑몰, 블로그, 검색 엔진 등 보여줄 콘텐츠가 여러 개인 서비스에서는 거의 필수로 이용.

9.1 프로젝트 구상

▣ **페이징이란..??**

- 게시물이 많을 경우 목록을 10~20개 정도씩 나눠 페이지별로 출력하는 기능

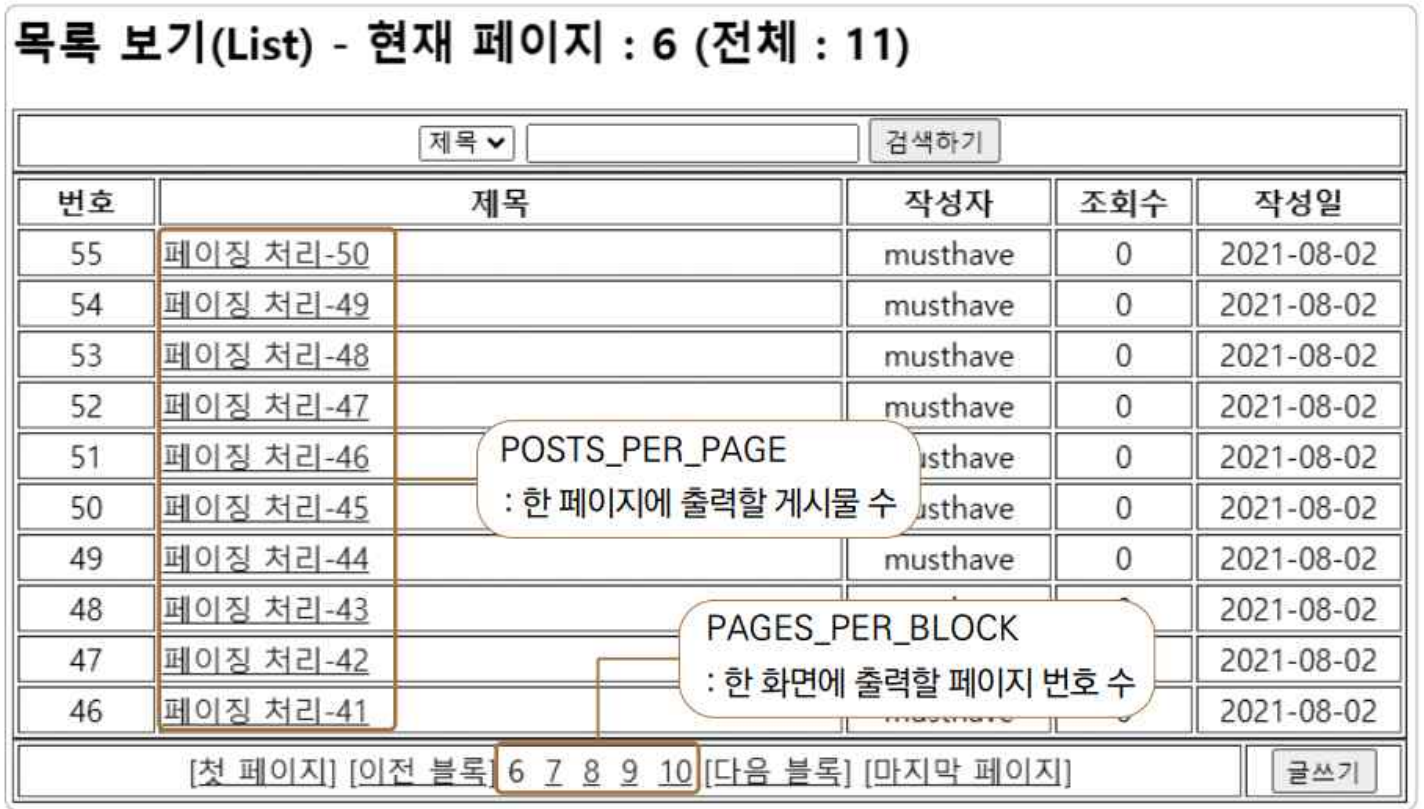
▣ **기능이 없을때의 문제점**

- 스크롤이 길어져서 사용자가 목록에서 원하는 게시글을 찾기 어려움
- 전송해야 할 데이터가 많아지므로 페이지 로딩 속도가 느려짐
- 한꺼번에 많은 데이터를 처리해야 하므로 데이터베이스에도 과부하가 걸림

▣ **페이징을 위한 설정**

- 한 페이지에 출력할 게시물의 개수
  - POSTS _PER_PAGE = 10
- 한 화면(블록)에 출력할 페이지 번호의 개수
  - PAGES_PER_BLOCK = 5

9.1 프로젝트 구상


목록 보기(List) - 현재 페이지 : 6 (전체 : 11)
제목
검색하기
번호
제목
작성자
조회수
작성일
55 페이징 처리-50 musthave 0 2021-08-02
54 페이징 처리-49 musthave 0 2021-08-02
53 페이징 처리-48 musthave 0 2021-08-02
52 페이징 처리-47 musthave 0 2021-08-02
51 페이징 처리-46 musthave 0 2021-08-02
50 페이징 처리-45 musthave 0 2021-08-02
49 페이징 처리-44 musthave 0 2021-08-02
48 페이징 처리-43 2021-08-02
47 페이징 처리-42 2021-08-02
46 페이징 처리-41 2021-08-02
POSTS_PER_PAGE
: 한 페이지에 출력할 게시물 수
PAGES_PER_BLOCK
: 한 화면에 출력할 페이지 번호 수
[첫 페이지] [이전 블록] 6 7 8 9 10 [다음 블록] [마지막 페이지]
글쓰기

9.1 프로젝트 구상

▣ 페이징 구현 절차

- 1단계 : board 테이블에 저장된 전체 레코드 수를 카운트
  - 전체 게시물이 105개라 가정
- 2단계 : 각 페이지에서 출력할 게시물의 범위를 계산
  - 계산식
    - 범위의 시작값 : (현재 페이지 - 1) * POSTS_PER_PAGE + 1
    - 범위의 종료값 : (현재 페이지 * POSTS_PER_PAGE)
  - 계산 예
    - 1페이지 일때
      - 시작값 : (1 - 1) * 10 + 1 = 1
      - 종료값 : 1 * 10 = 10
    - 2페이지 일때
      - 시작값 : (2 - 1) * 10 + 1 = 11
      - 종료값 : 2 * 10 = 20

9.1 프로젝트 구상

▣ 페이징 구현 절차(계속)

- 3단계 : 전체 페이지 수를 계산
  - 계산식 : Math.ceil(전체 게시물 수 / POSTS_PER_PAGE)
    - 계산 결과를 무조건 올림 처리해야 마지막 페이지를 조회할 수 있음
  - 계산 예
    - 게시물 수가 총 105개이므로
      - 페이지 수 : Math.ceil(105 / 10) = Math.ceil(10.5) = 11
- 4단계 : '이전 페이지 블록 바로가기'를 출력
  - 계산식 : ((현재 페이지 - 1) / PAGES_PER_BLOCK) * PAGES_PER_BLOCK + 1
    - 계산 예
      - 현재 1페이지일 때
        - pageTemp = ((1 - 1) / 5) * 5 + 1 = 1
      - 현재 5페이지일 때
        - pageTemp = ((5 - 1) / 5) * 5 + 1 = 1
    - pageTemp가 1이라면 첫번째 블록이므로 이전 블록 바로가기를 출력하지 않음

9.1 프로젝트 구상

▣ 페이징 구현 절차(계속)

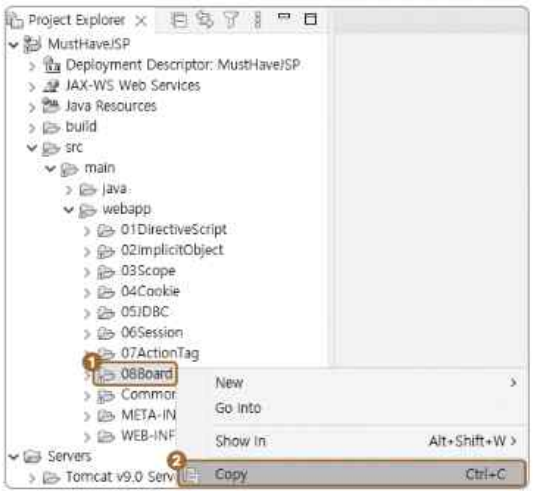
- 4단계 : (앞에서 계속)
  - 현재 6페이지일 때
    - pageTemp = ((6 - 1) / 5) * 5 + 1 = 6
  - 현재 10페이지일 때
    - pageTemp = ((10 - 1) / 5) * 5 + 1 = 6
  - 1이 아닐 때는 pageTemp - 1 결과로 이전 페이지 블록 바로가기를 출력
- 5단계 : 각 페이지 번호를 출력
  - 4단계에서 계산한 pageTemp를 BLOCK_PAGE만큼 반복하면서 +1 연산 후 출력
    - pageTemp 가 1일 때 : “1 2 3 4 5”를 출력
    - pageTemp 가 6일 때 : “6 7 8 9 10”을 출력
- 6단계 : ‘다음 페이지 블록 바로가기’를 출력
  - 각 페이지 번호를 출력한 후 pageTemp + 1 하여 다음 페이지 블록 바로가기를 설정

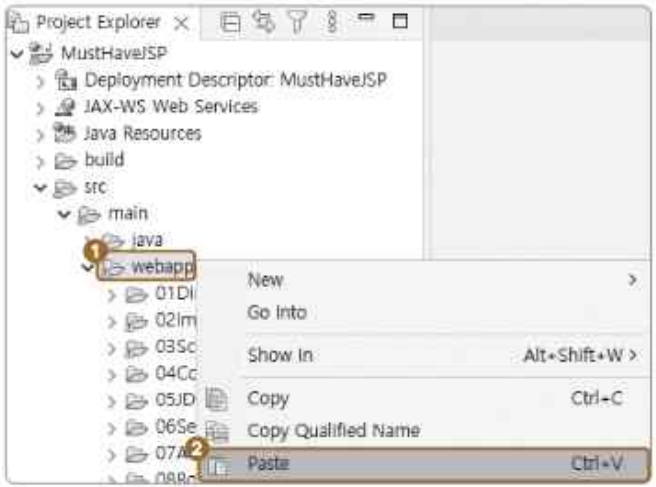
9.2 8장 소스 코드 복사

▣ 프로젝트에서 폴더 복사하기

. 08Board 폴더를 복사한 후 webapp에 붙여넣기

. 이름 충돌(Name Conflict) 창이 뜨면 09PagingBoard 로 폴더명 변경





9.3 더미 데이터 입력

▣ 페이징 테스트를 위한 더미데이터 100개 입력

. for문을 이용해서 insert 쿼리문을 100번 반복해서 실행

```
// DAO 객체를 통해 DB에 DTO 저장
BoardDAO dao = new BoardDAO(application);
//int iResult = dao.insertWrite(dto);  // 원래 코드 ❶
int iResult = 0;
for (int i = 1; i <= 100; i++) {
    dto.setTitle(title + "-" + i);  ❸
    iResult = dao.insertWrite(dto);
}
dao.close();
```

❷

예제 9-1] webapp/09PagingBoard/WriteProcess.jsp

❷ insert 쿼리문을 실행하는 메서드를 100번 반복

❸ “제목-1”, “제목-2”와 같이 입력되도록 수정

Write.jsp를 실행한 후 글 작성

9.4 페이징용 쿼리문 작성

▣ rownum 이란..??

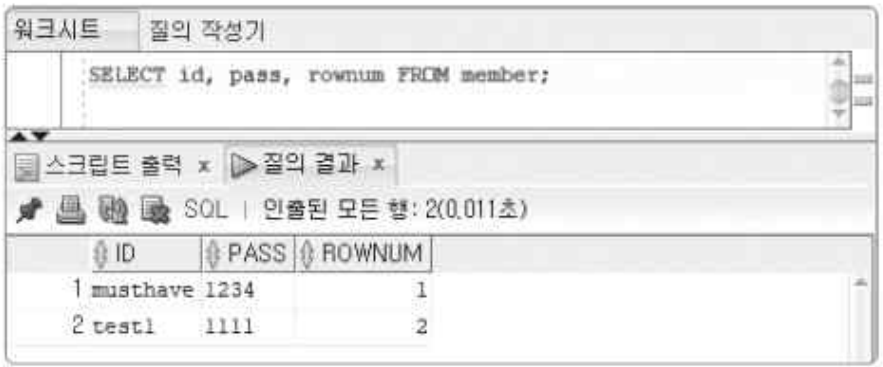
. 오라클에서 생성된 모든 테이블에서 사용할 수 있는 가상의 컬럼

. SELECT 쿼리문으로 추출 하는 데이터(row)에 순차적으로 부여되는 순번(num)

▣ SQL Developer 에서 member 테이블을 통해 rownum 확인

. member 테이블에 물리적으로 존재하지 않지만 select를 통해 출력됨

```
SELECT id, pass, rownum FROM member;
```



9.4 페이징용 쿼리문 작성

▣ 페이징 처리용 쿼리문 작성

. 첫 번째 페이지에 출력할 게시물을 가져오기 위해 rownum은 1~10까지로 지정

```
SELECT * FROM (  ❸
    SELECT Tb.*, rownum rNum FROM (  ❷
        SELECT * FROM board ORDER BY num DESC  ❶
    ) Tb
)
WHERE rNum BETWEEN 1 and 10;
```

❶ board 테이블의 게시물을 일련번호의 내림차순으로 정렬
❷ 정렬된 상태에서 rownum을 부여
❸ 순차적인 rownum을 통해 1~10까지의 게시물의 구간을 정해 가져올 수 있음

9.4 페이징용 쿼리문 작성

▣ 페이징 처리용 쿼리문 작성(계속)

. 만약 검색 조건을 추가해야 한다면 ❶에 where절을 삽입하면 됨

```
SELECT * FROM (
    SELECT Tb.*, rownum rNum FROM (
        SELECT * FROM board WHERE title LIKE '%검색어%' ORDER BY num DESC
    ) Tb
)
WHERE rNum BETWEEN 1 and 10;
```

. 검색조건 없이 쿼리문을 실행한 결과

스크립트 출력 × 질의 결과 ×

SQL | 인출된 모든 행: 10(0.029초)

	NUM	TITLE	CONTENT	ID	POSTDATE	VISITCOUNT
1	102	페이징 처리-100	페이징 처리 잘 되나요?	musthave	22/10/16	
2	101	페이징 처리-99	페이징 처리 잘 되나요?	musthave	22/10/16	
3	100	페이징 처리-98	페이징 처리 잘 되나요?	musthave	22/10/16	
4	99	페이징 처리-97	페이징 처리 잘 되나요?	musthave	22/10/16	
5	98	페이징 처리-96	페이징 처리 잘 되나요?	musthave	22/10/16	

9.5 DAO 수정

```
// 검색 조건에 맞는 게시물 목록을 반환합니다(페이징 기능 지원).
public List<BoardDTO> selectListPage(Map<String, Object> map) {
    List<BoardDTO> bbs = new Vector<BoardDTO>(); // 결과(게시물 목록)를 담을 변수

    // 쿼리문 템플릿 ❶
    String query = " SELECT * FROM ( "
                 + "    SELECT Tb.*, ROWNUM rNum FROM ( "
                 + "        SELECT * FROM board ";

    // 검색 조건 추가 ❷
    if (map.get("searchWord") != null) {
        query += " WHERE " + map.get("searchField")
               + " LIKE '%" + map.get("searchWord") + "%' ";
    }

    query += "      ORDER BY num DESC "
           + "    ) Tb "
           + " ) "
           + " WHERE rNum BETWEEN ? AND ?";  ❸
```

❶ 페이징 처리를 위한 쿼리문 작성
❷ 검색어가 있는 경우 where절 동적 추가
❸ 게시물의 구간을 설정하는 부분은 인파라미터로 작성

예제 9-2] Java Resources/model1/board/BoardDAO.java

9.5 DAO 수정

```
try {
    // 쿼리문 완성 ❹
    psmt = con.prepareStatement(query);
    psmt.setString(1, map.get("start").toString());
    psmt.setString(2, map.get("end").toString());

    // 쿼리문 실행 ❺
    rs = psmt.executeQuery();

    while (rs.next()) {
        // 한 행(게시물 하나)의 데이터를 DTO에 저장
        BoardDTO dto = new BoardDTO();
        dto.setNum(rs.getString("num"));
        dto.setTitle(rs.getString("title"));
        // 반환할 결과 목록에 게시물 추가
        bbs.add(dto);
    }
}
catch (Exception e) {
```

❹❺ 쿼리문의 인파라미터를 설정한 후 실행
반환된 ResultSet에 저장된 레코드를 List컬렉션에 추가

[Note] 기존 selectList()에서 쿼리문만 변경되어, 인파라미터를 설정하는 부분만 추가됨

예제 9-2] Java Resources/model1/board/BoardDAO.java(계속)

9.6 List.jsp 수정

▣ 설정값 관리

. web.xml에 컨텍스트 초기화 매개변수로 추가

예제 9-3] webapp/WEB-INF/web.xml

```
<context-param>
   <param-name>POSTS_PER_PAGE</param-name>
   <param-value>10</param-value>
</context-param>
<context-param>
   <param-name>PAGES_PER_BLOCK</param-name>
   <param-value>5</param-value>
</context-param>
```

. POSTS_PER_PAGE : 한 페이지에 출력할 게시물의 개수

. PAGES_PER_BLOCK : 한 화면에 출력할 페이지 번호의 개수

9.6 List.jsp 수정

▣ 데이터 계산

```
int totalCount = dao.selectCount(param);  // 게시물 수 확인

/*** 페이지 처리 start ***/
// 전체 페이지 수 계산 ❶
int pageSize = Integer.parseInt(application.getInitParameter("POSTS_PER_PAGE"));
int blockPage = Integer.parseInt(application.getInitParameter("PAGES_PER_BLOCK"));
int totalPage = (int)Math.ceil((double)totalCount / pageSize); // 전체 페이지 수

// 현재 페이지 확인 ❷
int pageNum = 1;  // 기본값
String pageTemp = request.getParameter("pageNum");
if (pageTemp != null && !pageTemp.equals(""))
    pageNum = Integer.parseInt(pageTemp); // 요청받은 페이지로 수정

// 목록에 출력할 게시물 범위 계산 ❸
int start = (pageNum - 1) * pageSize + 1;  // 첫 게시물 번호
int end = pageNum * pageSize; // 마지막 게시물 번호
param.put("start", start);
param.put("end", end);
/*** 페이지 처리 end ***/

List<BoardDTO> boardLists = dao.selectListPage(param);  // 게시물 목록 받기 ❹
```

❶ 컨텍스트 초기화 매개변수를 읽어와서 전체 페이지수 계산

❷ 파라미터로 전달된 현재 페이지 확인. 기본값은 1

❸ 쿼리문의 between절에 사용할 게시물의 구간 계산

[Note] 게시물의 수를 카운트하는 selectCount()와 게시물을 인출하는 selectListPage() 사이에 페이징 관련 코드를 삽입한다.

예제 9-4] webapp/09PagingBoard/List.jsp

9.6 List.jsp 수정

▣ 바로가기 **HTML** 코드 생성

```
int totalPages = (int) (Math.ceil(((double) totalCount / pageSize)));  ❶

// 단계 4 : '이전 페이지 블록 바로가기' 출력
int pageTemp = (((pageNum - 1) / blockPage) * blockPage) + 1;
if (pageTemp != 1) {  ❷
    pagingStr += "<a href='" + reqUrl + "?pageNum=1'>[첫 페이지]</a>";  ❸
    pagingStr += " ";
    pagingStr += "<a href='" + reqUrl + "?pageNum=" + (pageTemp - 1)
                 + "'>[이전 블록]</a>";  ❹
}
```

예제 9-5] Java Resources/utils/BoardPage.java (이전 페이지 블록 바로가기)

❶ 전체페이지수 계산

❷ 4단계에서 설명한 '이전 페이지 블록 바로가기'를 출력하기 위한 코드
pageTemp가 1이 아닐때만 '이전 블록' 링크를 출력

9.6 List.jsp 수정

▣ 바로가기 **HTML** 코드 생성(계속)

```
// 단계 5 : 각 페이지 번호 출력
int blockCount = 1;
while (blockCount <= blockPage && pageTemp <= totalPages) {
    if (pageTemp == pageNum) {  ❺
        // 현재 페이지는 링크를 걸지 않음
        pagingStr += " " + pageTemp + " ";
    } else {
        pagingStr += " <a href='" + reqUrl + "?pageNum=" + pageTemp
                  + "'>" + pageTemp + "</a> ";  ❻
    }
    pageTemp++;  ❼
    blockCount++;
}
```

예제 9-5] Java Resources/utils/BoardPage.java (각 페이지 번호 출력)

❺ 현재 페이지라면 링크를 걸지 않음
❻ 현재 페이지가 아닌 경우에만 링크를 삽입

9.6 List.jsp 수정

▣ 바로가기 **HTML** 코드 생성(계속)

```
        // 단계 6 : '다음 페이지 블록 바로가기' 출력
        if (pageTemp <= totalPages) {  ❽
            pagingStr += "<a href='" + reqUrl + "?pageNum=" + pageTemp
                       + "'>[다음 블록]</a>";  ❾
            pagingStr += " ";
            pagingStr += "<a href='" + reqUrl + "?pageNum=" + totalPages
                       + "'>[마지막 페이지]</a>";  ❿
        }

        return pagingStr;
    }
```

예제 9-5] Java Resources/utils/BoardPage.java (다음 페이지 블록 바로가기)

❽ pageTemp가 전체 페이지 수 이하일때 '다음 페이지 블록'을 출력

9.6 List.jsp 수정

▣ 화면출력

```
<body>
    <h2>목록 보기(List) - 현재 페이지 : <%= pageNum %> (전체 : <%= totalPage %>)
</h2>  ❶
else {
    // 게시물이 있을 때
    int virtualNumber = 0;  // 화면상에서의 게시물 번호
    int countNum = 0;
    for (BoardDTO dto : boardLists)
    {
        // virtualNum = totalCount--;  // 기존 코드
        virtualNum = totalCount - (((pageNum - 1) * pageSize) + countNum++);  ❷
%>
            <td>  ❸
                <%= BoardPage.pagingStr(totalCount, pageSize,
                        blockPage, pageNum, request.getRequestURI()) %>
            </td>
```

❶ 페이지 정보 출력

❷ 가상 게시물 번호에 현재 페이지 번호를 적용

❸ 각 페이지 바로가기 링크 출력

예제 9-6] webapp/09PagingBoard/List.jsp

9.7 동작 확인

▣ **List.jsp** 최초 실행

▣ **6**페이지로 이동한 경우

학습 마무리

▣ 핵심요약

- 게시판의 설정값은 web.xml에 컨텍스트 초기화 매개변수로 저장해 사용하면 소스 코드 수정 없이 값을 변경할 수 있음
- 해당 설정값과 테이블에 저장된 레코드의 개수를 통해 페이지 수를 계산
- 페이지 번호를 출력하는 코드는 14장에서 제작할 모델2 방식 게시판에서도 동일하게 사용
- 이처럼 공통적으로 쓰이는 기능은 별도의 유틸리티 클래스로 만들어두면 재사용성이 높아짐