

Part 2 기본적인 기능의 게시물 관리

Chapter 08 페이징 처리

- 페이징 처리의 핵심
 - 사용자에게 필요한 만큼의 데이터를 전송
 - 서버에서 최대한 빠르게 결과를 만들어 내야 한다
- 페이징 처리의 공부 단계
 - URI의 문자열을 조절해서 원하는 페이지의 데이터가 출력되도록 하는 단계
 - 목록 페이지 하단에 페이지 번호를 보여주고, 클릭하면 페이지가 이동하는 단계
 - 목록 페이지에서 조회나 수정 작업을 한 후에 다시 원래의 목록 페이지로 이동할 수 있게 처리하는 단계
- 페이징 처리 실행 단계
 - 1단계
 - 페이지 번호 등을 파라미터로 전달
 - SQL에 적절한 데이터를 뽑아서 보여줌
 - 2단계
 - 목록 페이지의 하단에 페이지 번호 출력
 - 클릭 시 원하는 페이지 이동
 - 3단계
 - 목록 페이지에서 특정 게시물 조회, 수정
 - 다시 원래 목록 페이지를 볼 수 있게 하는 기능

8.1 페이징 처리 방식

- JSP 내에서 페이징 처리 방식
 - 태그의 href 속성을 이용해서 직접 이동할 URI를 지정하는 방식
 - 태그를 이용해서 링크를 클릭하면 여러 정보를 전달하는 방식
 - 최소한의 정보를 이용하고 빠르게 개발할 수 있는 태그를 이용해서 처리

8.1.1 페이징 처리 원칙

- GET 방식만을 이용해서 처리
- 다른 사람에게 URL로 전달하는 경우가 많기 때문
- 페이징 처리가 되면 조회 화면에서 반드시 '목록 가기' 필요
- 목록 페이지에서 3페이지 보다가 특정 게시물을 보았다면, 다시 '목록 가기' 버튼을 눌러서 다시 3페이지의 목록 페이지로 이동하는 기능
- 페이징 처리에는 반드시 필요한 페이지 번호만을 제공

8.2 페이징 처리 개발에 필요한 지식

- 기술적인 내용
 - 페이징 처리를 위한 SQL
 - 데이터 개수 파악을 위한 SQL
 - 자바스크립트 혹은 태그를 통한 이벤트 처리
- 페이징 처리 개발 순서
 - 단순히 페이지 데이터가 화면에 출력하는 작업
 - 화면 하단에 페이지 번호가 표시 및 기능
 - 조회 페이지에서 목록 가기 선택 시 페이지 정보를 유지한 채로 이동하는 작업

8.2.1 MySQL의 limit를 이용한 페이지 출력 SQL

`limit 0, 10` -- 0번 부터 10개의 데이터를 가져온다.

8.2.1.1 충분한 양의 데이터 넣기

```
insert into tbl_board (  
    title  
    , content  
    , writer  
)(  
    SELECT  
        title  
        , content  
        , writer  
    FROM tbl_board  
);
```

8.3 MyBatis의 BoardDAO 처리

8.3.1 BoardDAO, XML Mapper, BoardDAOImpl 처리

- BoardDAO 인터페이스에 페이징 처리와 기능 추가

```
public interface BoardDAO {  
  
    public void create(BoardVO vo) throws Exception;  
  
    public BoardVO read(Integer bno) throws Exception;  
  
    public void update(BoardVO vo) throws Exception;  
  
    public void delete(Integer bno) throws Exception;  
  
    public List<BoardVO> listAll() throws Exception;
```

```
// 페이지 기능
public List<BoardVO> listPage(int page) throws Exception;

}
```

- boardMapper.xml

```
<select id="listPage" resultType="BoardVO">
  <![CDATA[
    select
      bno, title, content, writer, regdate, veiwcnt
    from
      tbl_board
    where bno > 0
    order by bno desc, regdate desc
    limit #{page}, 10
  ]]>
</select>
```

- BoardDAOImpl

```
@Override
public List<BoardVO> listPage(int page) throws Exception {
    if (page <= 0) {
        page = 1;
    }
    page = (page - 1) * 10;
    return session.selectList(namespace + ".listPage", page);
}
```

8.3.2 페이징 처리의 SQL 테스트

- BoardDAOTest

```
@Test
public void testListPage() throws Exception {
    int page = 3;
    List<BoardVO> list = dao.listPage(page);
    for (BoardVO boardVO : list) {
        logger.info(boardVO.getBno() + ":" + boardVO.getTitle());
    }
}
```

```

<terminated> BoardDAOTest [JUnit] C:\Program Files\Java\jre1.8.0_141\bin\javaw.exe (2018. 1. 27. 오후 10:46:56)
INFO : jdbc.resultset - 1. ResultSet.isNull() returned false
INFO : jdbc.resultset - 1. ResultSet.getString(title) returned title
INFO : jdbc.resultset - 1. ResultSet.isNull() returned false
INFO : jdbc.resultset - 1. ResultSet.getClob(content) returned com.mysql.jdbc.Clob@6574a52c
INFO : jdbc.resultset - 1. ResultSet.isNull() returned false
INFO : jdbc.resultset - 1. ResultSet.getString(writer) returned user00
INFO : jdbc.resultset - 1. ResultSet.isNull() returned false
INFO : jdbc.resultset - 1. ResultSet.getTimestamp(regdate) returned 2018-01-27 13:32:15.0
INFO : jdbc.resultset - 1. ResultSet.isNull() returned false
INFO : jdbc.resultset - 1. ResultSet.getInt(viewcnt) returned 0
INFO : jdbc.resultset - 1. ResultSet.isNull() returned false
INFO : jdbc.resultsettable -


| bno   | title | content  | writer | regdate               | viewcnt |
|-------|-------|----------|--------|-----------------------|---------|
| 27613 | title | [unread] | user00 | 2018-01-27 13:32:15.0 | 0       |
| 27612 | title | [unread] | user00 | 2018-01-27 13:32:15.0 | 0       |
| 27611 | 타이틀   | [unread] | user00 | 2018-01-27 13:32:15.0 | 0       |
| 27610 | title | [unread] | user00 | 2018-01-27 13:32:15.0 | 0       |
| 27609 | title | [unread] | user00 | 2018-01-27 13:32:15.0 | 0       |
| 27608 | Title | [unread] | user00 | 2018-01-27 13:32:15.0 | 0       |
| 27607 | Title | [unread] | user00 | 2018-01-27 13:32:15.0 | 0       |
| 27606 | title | [unread] | user00 | 2018-01-27 13:32:15.0 | 0       |
| 27605 | title | [unread] | user00 | 2018-01-27 13:32:15.0 | 0       |
| 27604 | title | [unread] | user00 | 2018-01-27 13:32:15.0 | 0       |


INFO : jdbc.resultset - 1. ResultSet.next() returned false
INFO : jdbc.resultset - 1. ResultSet.close() returned void
INFO : jdbc.audit - 1. PreparedStatement.getConnection() returned net.sf.log4jdbc.sql.jdbcapi.ConnectionSpy@20140db9
INFO : jdbc.audit - 1. Connection.getMetaData() returned com.mysql.jdbc.JDBC4DatabaseMetaData@1e6a3214
INFO : jdbc.audit - 1. PreparedStatement.getMoreResults() returned false
INFO : jdbc.audit - 1. PreparedStatement.getUpdateCount() returned -1
INFO : jdbc.audit - 1. PreparedStatement.close() returned
INFO : jdbc.connection - 1. Connection closed
INFO : jdbc.audit - 1. Connection.close() returned
INFO : org.zerock.test.BoardDAOTest - 27613:title
INFO : org.zerock.test.BoardDAOTest - 27612:title
INFO : org.zerock.test.BoardDAOTest - 27611:타이틀
INFO : org.zerock.test.BoardDAOTest - 27610:title
INFO : org.zerock.test.BoardDAOTest - 27609:title
INFO : org.zerock.test.BoardDAOTest - 27608>Title
INFO : org.zerock.test.BoardDAOTest - 27607>Title
INFO : org.zerock.test.BoardDAOTest - 27606:title
INFO : org.zerock.test.BoardDAOTest - 27605:title
INFO : org.zerock.test.BoardDAOTest - 27604:title
INFO : org.springframework.context.support.GenericApplicationContext - Closing org.springframework.context.support.GenericApplicat

```

- SQL refactoring 요소
 - 한 페이지에서 보여지는 데이터가 10개가 아니라면 limit 구문의 마지막 사용되는 10이라는 숫자 역시 변경되어야한다.
 - 매번 원하는 페이지를 처리할 때마다 계산을 해야 함
- SQL 해결책
 - 파라미터 2개를 받도록 한다.

8.4 DAO 처리를 도와줄 Criteria 클래스 만들기

```

public class Criteria {

    private int page;
    private int perPageNum;

    public Criteria() {
        this.page = 1;
        this.perPageNum = 10;
    }

    public int getPage() {
        return page;
    }

    public void setPage(int page) {

        if (page <= 0) {

```

```

        this.page = 1;
        return ;
    }
    this.page = page;
}
// method for MyBatis SQL Mapper
public int getPerPageNum() {
    return this.perPageNum;
}

public void setPerPageNum(int perPageNum) {
    if(perPageNum <= 0 || perPageNum > 100) {
        this.perPageNum = 10;
        return;
    }
    this.perPageNum = perPageNum;
}

// method for MyBatis SQL Mapper
public int getPageStart() {
    return (this.page - 1) * perPageNum;
}

@Override
public String toString() {
    return "Criteria [page=" + page + ", perPageNum=" + perPageNum + "];"
}
}

```

- 기본 값으로 페이지 번호 1페이지, 리스트당 데이터의 수는 10으로 강제 부여
- getPageStart()는 limit 구문에서 시작 위치를 지정할 때 사용
 - 시작 데이터 번호 = (페이지 번호 - 1) * 페이지 당 보여지는 개수

8.4.1 BoardDAO 인터페이스의 수정

```

public interface BoardDAO {

    public void create(BoardVO vo) throws Exception;

    public BoardVO read(Integer bno) throws Exception;

    public void update(BoardVO vo) throws Exception;

    public void delete(Integer bno) throws Exception;

    public List<BoardVO> listAll() throws Exception;

    public List<BoardVO> listPage(int page) throws Exception;

    public List<BoardVO> listCriteria(Criteria cri) throws Exception;
}

```

```
}
```

8.4.2 XML Mapper의 수정

- boardMapper.xml

```
<select id="listCriteria" resultType="BoardVO">
<![CDATA[
  SELECT
    bno, title, content, writer, regdate, viewcnt
  FROM
    tbl_board
  WHERE bno > 0
  ORDER BY bno DESC, regdate DESC
  LIMIT #{pageStart}, #{perPageNum};
]]>
</select>
```

8.4.3 BoardDAOImpl의 수정

```
@Override
public List<BoardVO> listCriteria(Criteria cri) throws Exception {
    return session.selectOne(namespace + ".listCriteria", cri);
}
```

8.4.4 BoardDAOTest에서의 테스트 작업

```
@Test
public void testListCriteria() throws Exception {
    Criteria criteria = new Criteria();
    criteria.setPage(2);
    criteria.setPerPageNum(20);

    List<BoardVO> list = dao.listCriteria(criteria);
    for (BoardVO boardVO : list) {
        logger.info(boardVO.getBno() + ":" + boardVO.getTitle());
    }
}
```

```

INFO : jdbc.audit - 1. PreparedStatement.executeUpdateCount()
INFO : jdbc.audit - 1. PreparedStatement.close() returned
INFO : jdbc.connection - 1. Connection closed
INFO : jdbc.audit - 1. Connection.close() returned
INFO : org.zerock.test.BoardDAOTest - 221146:title
INFO : org.zerock.test.BoardDAOTest - 221145:title
INFO : org.zerock.test.BoardDAOTest - 221144:타이틀
INFO : org.zerock.test.BoardDAOTest - 221143:title
INFO : org.zerock.test.BoardDAOTest - 221142:title
INFO : org.zerock.test.BoardDAOTest - 221141:Title
INFO : org.zerock.test.BoardDAOTest - 221140:Title
INFO : org.zerock.test.BoardDAOTest - 221139:title
INFO : org.zerock.test.BoardDAOTest - 221138:title
INFO : org.zerock.test.BoardDAOTest - 221137:title
INFO : org.zerock.test.BoardDAOTest - 221136:title
INFO : org.zerock.test.BoardDAOTest - 221135:title
INFO : org.zerock.test.BoardDAOTest - 221134:타이틀
INFO : org.zerock.test.BoardDAOTest - 221133:타이틀
INFO : org.zerock.test.BoardDAOTest - 221132:title
INFO : org.zerock.test.BoardDAOTest - 221131:title
INFO : org.zerock.test.BoardDAOTest - 221130:Title
INFO : org.zerock.test.BoardDAOTest - 221129:Title
INFO : org.zerock.test.BoardDAOTest - 221128:title
INFO : org.zerock.test.BoardDAOTest - 221127:title

```

👉 bno	title	content	writer	regdate	viewcnt
221,146	title	content	user00	2018-01-27 14:23:51	0
221,145	title	content	user00	2018-01-27 14:23:51	0
221,144	타이틀	content	user00	2018-01-27 14:23:51	0
221,143	title	content	user00	2018-01-27 14:23:51	0
221,142	title	content	user00	2018-01-27 14:23:51	0
221,141	Title	content	user00	2018-01-27 14:23:51	0
221,140	Title	content	user00	2018-01-27 14:23:51	0
221,139	title	content	user00	2018-01-27 14:23:51	0
221,138	title	content	user00	2018-01-27 14:23:51	0
221,137	title	content	user00	2018-01-27 14:23:51	0
221,136	title	content	user00	2018-01-27 14:23:51	0
221,135	title	content	user00	2018-01-27 14:23:51	0
221,134	title	content	user00	2018-01-27 14:23:51	0
221,133	타이틀	content	user00	2018-01-27 14:23:51	0
221,132	title	content	user00	2018-01-27 14:23:51	0
221,131	title	content	user00	2018-01-27 14:23:51	0
221,130	Title	content	user00	2018-01-27 14:23:51	0
221,129	Title	content	user00	2018-01-27 14:23:51	0
221,128	title	content	user00	2018-01-27 14:23:51	0
221,127	title	content	user00	2018-01-27 14:23:51	0

8.5 BoardService 수정하기

- BoardController와 BoardDAO와의 연결 작업을 담당하는 BoardService

```

public interface BoardService {

    public void regist(BoardVO board) throws Exception;

    public BoardVO read(Integer bno) throws Exception;

    public void modify(BoardVO board) throws Exception;

    public void remove(Integer bno) throws Exception;

    public List<BoardVO> listAll() throws Exception;

    public List<BoardVO> listCriteria(Criteria criteria) throws Exception;

}

```

- BoardServiceImpl

```

@Override
public List<BoardVO> listCriteria(Criteria cri) throws Exception {
    return dao.listCriteria(cri);
}

```