

Spring Framework

- 게시판
- 1. 구현 순서
- 2. 페이징 처리

1. 프로젝트 준비사항

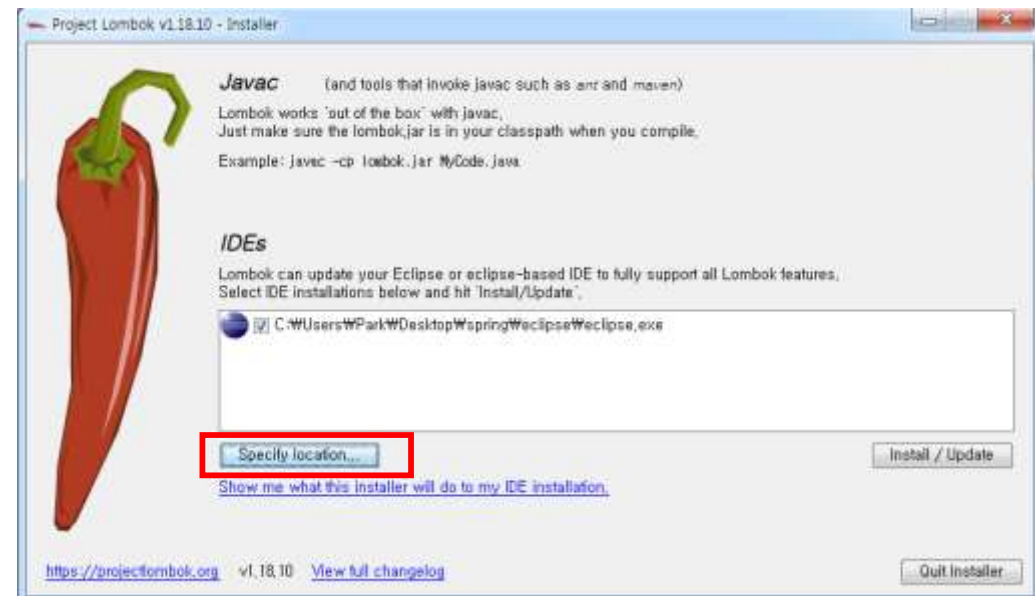
Lombok라이브러리 설치

- 스프링 플러그인 만으로 개발이 가능하지만. Lombok을 이용하면 `getter/setter`, `toString()`, 생성자를 빠르게 생성해줍니다.
- 실업무에서도 사용됩니다.
- 단, 모든 팀원이 롬복을 사용해야 하는 단점이 존재합니다.

다운로드 경로

<https://projectlombok.org/download>

1. Lombok.jar파일을 클릭해서 설치합니다.
또는
2. `java -jar lombok.jar` 명령어로 실행합니다
3. 설치가 자동으로 안된다면 이클립스 경로를 직접지정해서 설치 합니다
4. 설치가 완료되면 이클립스에 우측과 같은 파일이 생성됩니다



configuration	2019-11-04 오후...	파일 폴더	
dropins	2019-09-17 오전...	파일 폴더	
features	2019-11-03 오후...	파일 폴더	
p2	2019-11-04 오후...	파일 폴더	
plugins	2019-11-03 오후...	파일 폴더	
readme	2019-11-02 오후...	파일 폴더	
.eclipseproduct	2019-09-16 오전...	ECLIPSEPRODUCT...	1KB
artifacts.xml	2019-11-03 오후...	XML 문서	325KB
eclipse.exe	2019-11-02 오후...	응용 프로그램	416KB
eclipse.ini	2019-11-05 오전...	구성 설정	1KB
eclipsec.exe	2019-11-02 오후...	응용 프로그램	128KB
lombok.jar	2019-11-05 오전...	Executable Jar File	1,723KB

롬복 라이브러리 추가

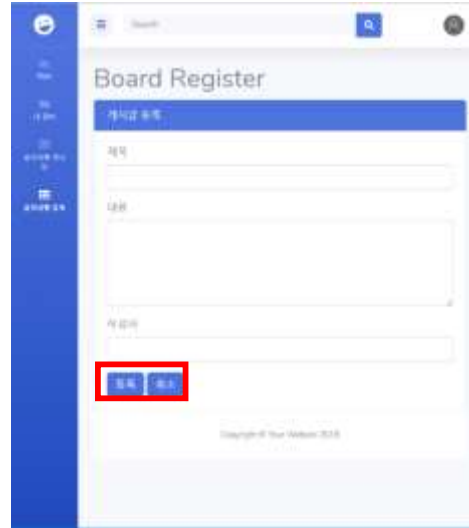
```
<dependency>
  <groupId>org.projectlombok</groupId>
  <artifactId>lombok</artifactId>
  <version>1.18.10</version>
  <scope>provided</scope>
</dependency>
```

1. 게시판 구현 순서



Board List Page showing a table of board entries. The 'Add' button is highlighted in red.

번호	제목	작성자	작성일	수정일
30	게시문	게시문	2019-04-22 06:01:00	2019-04-22
29	게시문	게시문	2019-04-22 06:01:01	2019-04-22
28	게시문	게시문	2019-04-22 06:01:01	2019-04-22
27	게시문	게시문	2019-04-22 06:01:01	2019-04-22
26	게시문	게시문	2019-04-22 06:01:01	2019-04-22
25	게시문	게시문	2019-04-22 06:01:01	2019-04-22
24	게시문	게시문	2019-04-22 06:01:01	2019-04-22
23	게시문	게시문	2019-04-22 06:01:01	2019-04-22
22	게시문	게시문	2019-04-22 06:01:01	2019-04-22
21	게시문	게시문	2019-04-22 06:01:01	2019-04-22



Board Register Page showing the form for adding a new board entry. The 'Add' button is highlighted in red.

Board Register

게시문 등록

제목

내용

작성자

등록



Board Read Page showing the details of a selected board entry. The 'Edit' button is highlighted in red.

Board Read Page

게시문 보기

제목

내용

작성자

등록

수정



Board Modify Page showing the form for editing a board entry. The 'Edit' button is highlighted in red.

Board Modify Page

수정 게시글

제목

내용

작성자

등록

수정

1. 게시판 구현 순서

1. 컨트롤러 생성(화면 확인)
 2. 등록 처리
 3. 테이블 생성
 4. DB관련 설정(root-xml작업)
 5. BoardVO 생성(DB컬럼명과 반드시 동일하게 생성)
 6. Service구현
 7. DAO 구현
 8. 마이바티스 DB작업
-
9. 상세보기 처리
 10. 변경 처리
 11. 삭제 처리
 12. 페이징 처리



테이블 SQL

```
create table freeboard(  
    bno number(10,0),  
    title varchar(200) not null,  
    content varchar2(2000) not null,  
    writer varchar2(50) not null,  
    regdate date DEFAULT sysdate,  
    update date DEFAULT sysdate  
);  
alter table freeboard add CONSTRAINT freeboard_pk PRIMARY key (bno);  
create SEQUENCE freeboard_seq;
```

2. 페이징

1. 반드시 GET 방식으로만 처리한다
2. 이동할 때 페이지 번호 를가지고 다닌다
3. 페이징 처리하는 로직을 클래스로 분류한다
 - > Criteria클래스, PageVO클래스

2. 페이징 (Criteria 클래스)

-페이징을 처리하는 기준

```

public class Criteria {

    private int pageNum; //페이지 번호
    private int count; //몇개의 데이터를 보여줄건가

    public Criteria() {
        this.pageNum = 1;
        this.count = 10;
    }

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

    public int getPageStart() {
        return (pageNum - 1) * count;
    }
}

```

이하 게터 세터 생성

mysql의 limit함수 쿼리문

```

select *
  from tbl_board
 order by num desc
limit 0, 10;

```

← 1번부터 10개
데이터가 조회된다
(데이터시작, 데이터개수)

```

select *
  from tbl_board
 order by num desc
limit 10, 20;

```

← 11번부터 20개
데이터 가 조회된다

마이바티스의 sql문

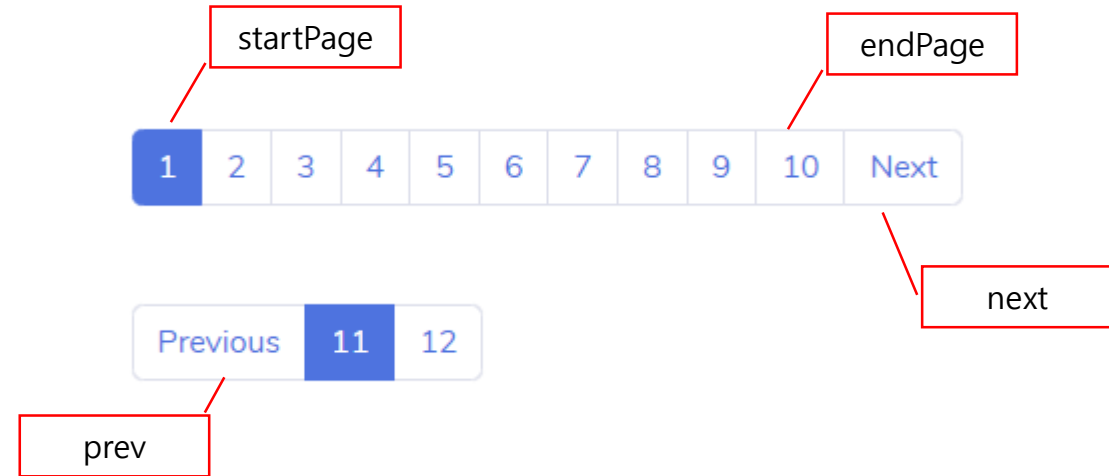
```

select *
  from tbl_board
 order by num desc
limit #{pageStart} , #{count}

```

2. 페이징 (PageVO 클래스) -페이징 계산 처리 클래스

- 1 total 게시판 글 전체 개수.
- 2 endPage: 게시판을 화면에 보여질 마지막 페이지 번호.
- 3 startPage: 게시판을 화면에 보여질 첫번째 페이지 번호.
- 4 realEnd: 게시판의 실제 마지막 페이지 번호.
- 5 prev: 이전 페이지 버튼 활성화 여부.
- 6 next: 다음 페이지 버튼 활성화 여부.



공식

2 endPage =

1~10 페이지 클릭시 10
11~20 페이지 클릭시 20
21~30 페이지 클릭시 30

3 startPage =

endPage가 10 일때 1
endPage가 20 일때 11
endPage가 30 일때 21

4 realEnd =

게시물이 50개 라면 5
게시물이 51개 라면 6
게시물이 101개 라면 11

공식

4 prev =

startPage가 1이면 비활성화
startPage가 11이면 활성화

5 next =

realEnd가 endPage보다 크면 활성화

Math.ceil() 함수를 이용한다

2. 페이징 (PageVO 클래스)

```
private int startPage; // 화면에 보여지는 시작 페이지 번호
private int endPage; // 화면에 보여지는 끝 페이지 번호
private boolean prev, next; //다음, 이전 버튼
private int total; //총 게시물 수
private Criteria cri; //기준

//생성자
public PageVO(Criteria cri, int total) {
    this.cri = cri; //기준
    this.total = total; //총 게시물 수
    this.endPage = (int)Math.ceil(cri.getPageNum() / (double)10 ) * 10;
    this.startPage = endPage - 10 + 1;
    int realEnd = (int)Math.ceil(total / (double)cri.getCount() );

    if(this.endPage > realEnd ) {
        this.endPage = realEnd;
    }

    this.prev = this.startPage > 1;
    this.next = realEnd > this.endPage;
}
//이후 아래 게터 세터
```


2. 검색과 페이징

자유게시판

게시판의 **검색기능**을 폼으로 사용된다

총 113게시글

제목

검색

번호	제목	작성자	등록일	수정일
141	ㅎㅎ	두글	2019-11-07 01:44:14.0	2019-11-07 01:44:14.0
131	admin2	test2	2019-11-05 18:56:11.0	2019-11-05 18:56:11.0
130	admin2	test2	2019-11-05 18:56:11.0	2019-11-05 18:56:11.0
129	admin2	test2	2019-11-05 18:56:11.0	2019-11-05 18:56:11.0
128	admin2	test2	2019-11-05 18:56:11.0	2019-11-05 18:56:11.0
127	admin2	test2	2019-11-05 18:56:11.0	2019-11-05 18:56:11.0
126	admin2	test2	2019-11-05 18:56:11.0	2019-11-05 18:56:11.0
125	admin2	test2	2019-11-05 18:56:11.0	2019-11-05 18:56:11.0
124	admin1	test1	2019-11-05 18:56:06.0	2019-11-05 18:56:06.0
123	admin1	test1	2019-11-05 18:56:06.0	2019-11-05 18:56:06.0

1

2

3

4

5

6

7

8

9

10

다음

글쓰기

1. Form에서 pageNum, amount, searchType, searchName을 동시에 넘긴다.
Criteria에 search키워드 search타입 변수 추가

1. 페이징 쿼리를 **동적쿼리**로 변경
2. 전체게시글 수 **동적쿼리**로 변경
3. 페이징 처리시 searchType, searchName을 넘겨준다.
5. **검색 후 화면에서 일어나는 문제해결**
3페이지에서 검색하면 3페이지로 이동되는 문제
(검색버튼 클릭시 pageNum은 1로 처리)

select박스에 검색 조건이 남게 처리한다
input박스에 검색 단어가 남게 처리한다

1. 검색폼, 페이징폼에서 pageNum, amount, searchType, searchName를 각각 전송한다
hidden을 이용

페이징폼

```
<input type="hidden" name="page" value="1">  
<input type="hidden" name="amount" value="${pageVO.amount }">
```

2. Criteria에 search키워드 search타입 변수 추가

```
public class Criteria {  
  
    private int pageNum;  
    private int amount;  
  
    private String searchName;  
    private String searchType;  
  
    public Criteria() {  
        this(1, 10);  
    }  
  
    public Criteria(int pageNum, int amount) {  
        this.pageNum = pageNum;  
        this.amount = amount;  
    }  
  
}
```

동적 쿼리란?

마이바티스의 태그로써 조건문을 써서 쿼리의 실행을 제어하는 방법입니다
jstl구문과 사용방법이 유사합니다.

마이바티스의 test="" 구문 안에 작성되는 값은 VO의 getter나 , map의 key값이 쓰입니다

대표적 태그의 종류

1. if
2. choose(when, otherwise)
3. foreach
4. include등

3. 페이징 쿼리를 동적쿼리로 변경 전체게시글 쿼리를 동적쿼리로 변경

```
<select id="getList" resultType="FreeBoardVO">
  <![CDATA[
    select *
    from (select rownum as rn, bno, writer, title, content, regdate, updatedate
          from (select *
                from freeboard
                where
                  <if test="searchType == 'title' ">title like '%' || #{searchName} || '%</if>
                  <if test="searchType == 'content' ">content like '%' || #{searchName} || '%</if>
                  <if test="searchType == 'writer' ">writer like '%' || #{searchName} || '%</if>
                  <if test="searchType == 'titcont' ">title like '%' || #{searchName} || '%' or content like '%' || #{searchName} || '%</if>
                  <if test="searchType == null or searchType == '' ">1=1</if>

          ) where rn > (#{pageNum}-1) * #{amount} and rn <= (#{pageNum}* #{amount})
        )
  ]]>
  order by bno desc
)
  ]]>
</select>
```

4. 화면에서 나타나는 에러들을 처리해준다.

스프링1부 끝 수고하셨습니다