





게시판 구현하기 (리스트)

게시판 구현하기 - 1

- 설명

MVC 패턴을 사용하기 위한 패키지 추가

```
▼  src/main/java
  >  kr.co.greenart.board.controller
  >  kr.co.greenart.board.model.dao
  >  kr.co.greenart.board.model.dto
  >  kr.co.greenart.board.model.service
  >  . . .
```

게시판 구현하기 - 2

• 설명

아래의 boardList.jsp 템플릿을 활용해 가져오기

```
1<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
2<%@ include file="/WEB-INF/include/header.jsp"%>
3<%@ include file="/WEB-INF/include/footer.jsp"%>
4<%@ include file="/WEB-INF/include/notice.jsp"%>
5<!-- 게시판 목록 -->
6<table border="1">
7<thead>
8<tr>
9<th>번호</th>
10<th>제목</th>
11<th>작성자</th>
12<th>작성일</th>
13<th>조회수</th>
14</tr>
15</thead>
16<tbody>
17<tr>
18<td>1</td>
19<td>제목 1</td>
20<td>작성자 1</td>
21<td>2023-01-01</td>
22<td>100</td>
23</tr>
24<tr>
25<td>2</td>
26<td>제목 2</td>
27<td>작성자 2</td>
28<td>2023-01-02</td>
29<td>200</td>
30</tr>
31<tr>
32<td>3</td>
33<td>제목 3</td>
34<td>작성자 3</td>
35<td>2023-01-03</td>
36<td>300</td>
37</tr>
38<tr>
39<td>4</td>
40<td>제목 4</td>
41<td>작성자 4</td>
42<td>2023-01-04</td>
43<td>400</td>
44</tr>
45<tr>
46<td>5</td>
47<td>제목 5</td>
48<td>작성자 5</td>
49<td>2023-01-05</td>
50<td>500</td>
51</tr>
52<tr>
53<td>6</td>
54<td>제목 6</td>
55<td>작성자 6</td>
56<td>2023-01-06</td>
57<td>600</td>
58</tr>
59<tr>
60<td>7</td>
61<td>제목 7</td>
62<td>작성자 7</td>
63<td>2023-01-07</td>
64<td>700</td>
65</tr>
66<tr>
67<td>8</td>
68<td>제목 8</td>
69<td>작성자 8</td>
70<td>2023-01-08</td>
71<td>800</td>
72</tr>
73<tr>
74<td>9</td>
75<td>제목 9</td>
76<td>작성자 9</td>
77<td>2023-01-09</td>
78<td>900</td>
79</tr>
80<tr>
81<td>10</td>
82<td>제목 10</td>
83<td>작성자 10</td>
84<td>2023-01-10</td>
85<td>1000</td>
86</tr>
87</tbody>
88</table>
89</pre>
```

게시판 구현하기 - 3

- 설명

co.kr.greenart.model.dto 패키지에 Board 클래스를 추가하고 getter, setter 생성

```
public class Board {  
    private int idx;  
    private String title;  
    private String content;  
    private String writer;  
    private String indate;  
    private int count;  
}
```

게시판 구현하기 - 4

- 설명

co.kr.greenart.model.service 패키지에 BoardService 인터페이스와 BoardServiceImpl 클래스 생성

- 설명

co.kr.greenart.model.dao 패키지에 BoardDao 클래스 생성

게시판 구현하기 - 5

• 설명

```
@Controller  
@RequestMapping("/board")  
public class BoardController {
```

해당 클래스를 스프링의 컨트롤러로 지정하고, 웹 요청을 처리하는 역할을 수행하기 위한 어노테이션

```
    @GetMapping("/list.do")  
    public String boardList(Model model) {  
        System.out.println("asdasd");  
        BoardDTO bo = new BoardDTO();
```

해당하는 GET요청 URL을 처리하기 위해 아래 코드(메서드)를 실행

모델 인터페이스를 통해 데이터를 추가하여 뷰에게 전달할 데이터를 저장 및 전달해주는 역할

```
        bo.setIdx(1);  
        bo.setTitle("게시판 제목");  
        bo.setContent("게시판 내용");  
        bo.setWriter("김재섭");  
        bo.setIndate("2023-01-01");  
        bo.setCount(13);  
  
        List<BoardDTO> list = new ArrayList<BoardDTO>();  
        list.add(bo);  
        list.add(bo);  
        list.add(bo);  
        list.add(bo);
```

```
        model.addAttribute("list", list); // 객체 바인딩  
        return "board/boardList"; // /WEB-INF/views/boardList.jsp 파일로 포워딩
```

Model에 객체를 바인딩
* 뷰에서는 바인딩 된 객체를 꺼내 사용 가능

↓
해당 경로를 알아서 찾을 수 있는 이유
servlet-context.xml에 경로와 파일명이 정의되어있음

게시판 구현하기 - 6

- 어노테이션(Annotation)

사전적 의미는 주석이라는 뜻이며, 특별한 의미 또는 기능을 수행하도록 선언하는 방식

- 참고

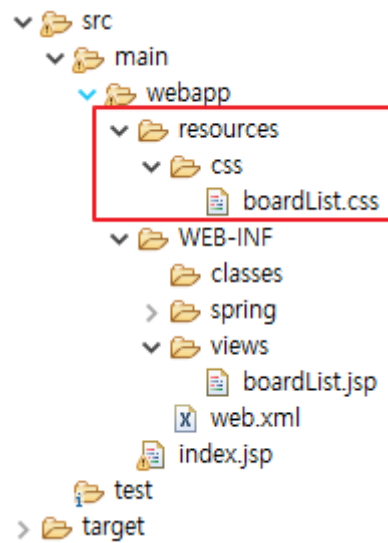
@Component는 초기 Spring에서 사용하던 어노테이션으로 Controller와 같은 의미이나, 가독성을 위해 주로 Controller를 사용함

폴더 구조 관리하기 (css)

폴더 구조 관리하기 - CSS

- 설명

resources/css/boardList.css 파일 추가



폴더 구조 관리하기 - CSS

• 설명

1. boardList.jsp 파일의 head 부분 style 안에 있는 내용 복사
2. resources/css/boardList.css 파일에 붙여 넣기 후 boardList.jsp style 영역 전부 삭제

```
<style>
.bd-placeholder-img {
  font-size: 1.125rem;
  text-anchor: middle;
  -webkit-user-select: none;
  -moz-user-select: none;
  user-select: none;
}

@media (min-width: 768px) {
  .bd-placeholder-img-lg {
    font-size: 3.5rem;
  }
}

.b-example-divider {
  height: 3rem;
  background-color: rgba(0, 0, 0, .1);
  border: solid rgba(0, 0, 0, .15);
  border-width: 1px 0;
  box-shadow: inset 0 .5em 1.5em rgba(0, 0, 0, .1);
}

.b-example-vr {
  flex-shrink: 0;
  width: 1.5rem;
}
```

폴더 구조 관리하기 - CSS

- 설명

boardList.jsp 파일의 head 부분에 link 추가

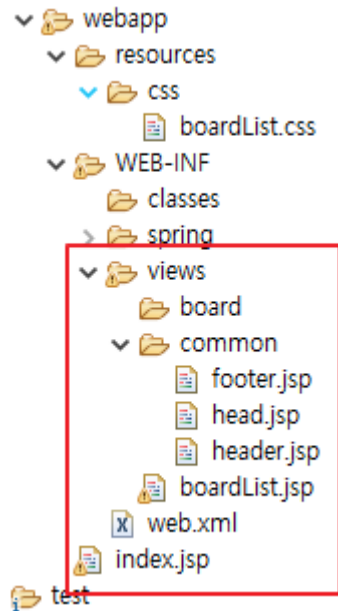
```
<!-- boardList.css 파일 불러오기 -->  
<link href="resources/css/boardList.css" rel="stylesheet">  
  
</head>
```

폴더 구조 관리하기 (jsp)

폴더 구조 관리하기 - jsp

• 설명

1. views 아래에 board, common 폴더 생성
2. common 폴더 아래에 footer, head, header jsp 파일 생성



폴더 구조 관리하기 - jsp

• 설명

1. boardList.jsp 파일의 head 부분을 모두 복사
2. common/head.jsp 파일에 붙여넣기
3. boardList.jsp 파일의 head 부분 삭제

boardList.jsp

```
<html lang="ko" class="h-100">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <meta name="description" content="">
  <meta name="author" content="Mark Otto, Jacob Thornton, 그리고 Bootstrap 기여자들">
  <meta name="generator" content="Hugo 0.104.2">
  <title>Sticky Footer Navbar Template · Bootstrap v5.2</title>

  <link rel="canonical" href="https://getbootstrap.kr/docs/5.2/examples/sticky-footer-navbar/">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" rel="stylesheet" integr

  <!-- Favicons -->
  <link rel="apple-touch-icon" href="/docs/5.2/assets/img/favicons/apple-touch-icon.png" sizes="180x180">
  <link rel="icon" href="/docs/5.2/assets/img/favicons/favicon-32x32.png" sizes="32x32" type="image/png">
  <link rel="icon" href="/docs/5.2/assets/img/favicons/favicon-16x16.png" sizes="16x16" type="image/png">
  <link rel="manifest" href="/docs/5.2/assets/img/favicons/manifest.json">
  <link rel="mask-icon" href="/docs/5.2/assets/img/favicons/safari-pinned-tab.svg" color="#712cf9">
  <link rel="icon" href="/docs/5.2/assets/img/favicons/favicon.ico">
  <meta name="theme-color" content="#712cf9">
  <!-- boardList.css 파일 불러오기 -->
  <link href="resources/css/boardList.css" rel="stylesheet">
</head>
<body class="d-flex flex-column h-100">
  <header>
    <!-- Fixed navbar -->
```

head.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <meta name="description" content="">
  <meta name="author" content="Mark Otto, Jacob Thornton, 그리고 Bootstrap 기여자들">
  <meta name="generator" content="Hugo 0.104.2">
  <title>Sticky Footer Navbar Template · Bootstrap v5.2</title>

  <link rel="canonical" href="https://getbootstrap.kr/docs/5.2/examples/stic
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstra

  <!-- Favicons -->
  <link rel="apple-touch-icon" href="/docs/5.2/assets/img/favicons/apple-touc
  <link rel="icon" href="/docs/5.2/assets/img/favicons/favicon-32x32.png" siz
  <link rel="icon" href="/docs/5.2/assets/img/favicons/favicon-16x16.png" siz
  <link rel="manifest" href="/docs/5.2/assets/img/favicons/manifest.json">
  <link rel="mask-icon" href="/docs/5.2/assets/img/favicons/safari-pinned-tab
  <link rel="icon" href="/docs/5.2/assets/img/favicons/favicon.ico">
  <meta name="theme-color" content="#712cf9">
  <!-- boardList.css 파일 불러오기 -->
  <link href="resources/css/boardList.css" rel="stylesheet">
</head>
```

폴더 구조 관리하기 - jsp

• 설명

1. boardList.jsp 파일의 header 부분을 모두 복사
2. common/header.jsp 파일에 붙여넣기
3. boardList.jsp 파일의 header 부분 삭제

boardList.jsp

```
<body class="d-flex flex-column h-100">
<header>
  <!-- Fixed navbar -->
  <nav class="navbar navbar-expand-md navbar-dark fixed-top bg-dark">
    <div class="container-fluid">
      <a class="navbar-brand" href="#">Fixed navbar</a>
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarCollapse">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarCollapse">
        <ul class="navbar-nav me-auto mb-2 mb-md-0">
          <li class="nav-item">
            <a class="nav-link active" aria-current="page" href="#">Home</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="#">Link</a>
          </li>
          <li class="nav-item">
            <a class="nav-link disabled">Disabled</a>
          </li>
        </ul>
        <form class="d-flex" role="search">
          <input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">
          <button class="btn btn-outline-success" type="submit">Search</button>
        </form>
      </div>
    </div>
  </nav>
</header>
<!-- Begin page content -->
<main class="flex-shrink-0">
```

header.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<header>
  <!-- Fixed navbar -->
  <nav class="navbar navbar-expand-md navbar-dark fixed-top bg-dark">
    <div class="container-fluid">
      <a class="navbar-brand" href="#">Fixed navbar</a>
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarCollapse">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarCollapse">
        <ul class="navbar-nav me-auto mb-2 mb-md-0">
          <li class="nav-item">
            <a class="nav-link active" aria-current="page" href="#">Home</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="#">Link</a>
          </li>
          <li class="nav-item">
            <a class="nav-link disabled">Disabled</a>
          </li>
        </ul>
        <form class="d-flex" role="search">
          <input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">
          <button class="btn btn-outline-success" type="submit">Search</button>
        </form>
      </div>
    </div>
  </nav>
</header>
```

폴더 구조 관리하기 - jsp

• 설명

1. boardList.jsp 파일의 footer 부분을 모두 복사 (부트스트랩 script까지)
2. common/footer.jsp 파일에 붙여넣기
3. boardList.jsp 파일의 footer 부분 삭제

boardList.jsp

```
</div>
</main>
<footer class="footer mt-auto py-3 bg-light">
  <div class="container">
    <span class="text-muted">Place sticky footer content here.</span>
  </div>
</footer>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.
</body>
</html>
```

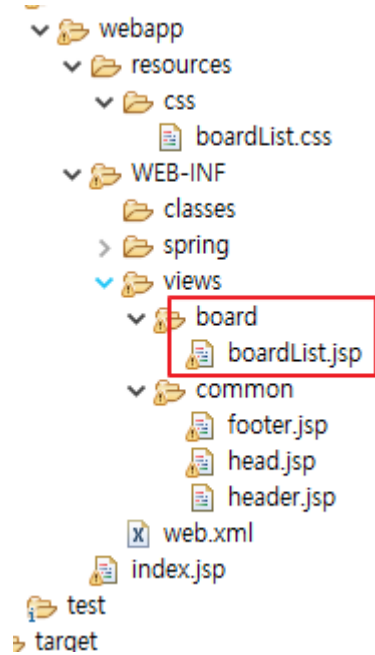
footer.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<footer class="footer mt-auto py-3 bg-light">
  <div class="container">
    <span class="text-muted">Place sticky footer content here.</span>
  </div>
</footer>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap
```


폴더 구조 관리하기 - jsp

- 설명

boardList.jsp 파일을 board 폴더 아래로 이동



폴더 구조 관리하기 - jsp

• 설명

1. boardController의 return 경로 변경
2. boardList.jsp에서 지웠던 부분에 include file 설정

boardController

```
model.addAttribute("list", |  
return "board/boardList"; /.
```

boardList.jsp

```
<!doctype html>  
<html lang="ko" class="h-100">  
  <%@ include file="../common/head.jsp" %>  
  
  <body class="d-flex flex-column h-100">  
    <%@ include file="../common/header.jsp" %>  
  
    <!-- Begin page content -->  
    <main class="flex-shrink-0">  
      <div class="container">  
  
        </ul>  
      </nav>  
    </div>  
  </main>  
  <%@ include file="../common/footer.jsp" %>  
</body>  
</html>
```

폴더 구조 관리하기 - jsp

- 폴더 구조를 관리하는 이유

1. 각각의 폴더에 있는 파일들의 역할이 분명해짐
2. 하나의 파일에 있던 코드가 줄어들어 필요한 코드만 찾아서 확인 가능
3. 공통적으로 사용되는 파일(common)에 수정사항이 생길 경우 해당되는 모든 페이지를 수정 하는것이 아니라, 해당되는 파일 하나만 수정하면 되므로 유지보수할 때 좋음

간단한 게시판 구현하기 (JSP)

게시판 구현하기 - JSP

- 설명

JSTL Core 라이브러리 사용을 위해 JSP 파일 상단에 아래 코드 작성

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>  
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
```

- JSTL (JavaServer Page Standard Tag Library)

예전에 사용하던 스크립틀릿의 가독성을 높여 사용하기 위해 JSTL을 씀

스크립틀릿 : <%=if %> , <%=for %> 등

JSTL : <c:if>, <c:forEach>

게시판 구현하기 - JSP

• 설명

컨트롤러에게 전달받은 list 객체가 비어있을 경우
"등록된 글이 없습니다." 출력

```
<c:choose>
  <c:when test="${empty list}">
    <tr>
      <td colspan="5"><h3 class="text-center">등록된 글이 없습니다.</h3></td>
    </tr>
  </c:when>
  <c:otherwise>
    <c:forEach var="item" items="${list}">
      <tr>
        <td>${item.idx}</td>
        <td>${item.title}</td>
        <td>${item.writer}</td>
        <td>${item.indate}</td>
        <td>${item.count}</td>
      </tr>
    </c:forEach>
  </c:otherwise>
</c:choose>
```

list 객체를 하나씩 꺼내고, 꺼낸 객체는 item이라는 이름으로 사용함

게시판 구현하기 - JSP

• 최종 화면

게시판

글쓰기

번호	제목	작성자	작성일	조회수
1	게시판 제목	김재섭	2023-01-01	13
2	게시판 제목	김재섭	2023-01-01	13
3	게시판 제목	김재섭	2023-01-01	13
4	게시판 제목	김재섭	2023-01-01	13

«12345»

게시판 구현하기

(MVC + MyBatis 연결)

게시판 구현하기 – MVC + ByBatis 연결

• BoardController

외부에서 객체를 생성하고 주입하는 어노테이션
개발자가 직접 서비스 객체를 생성할 필요가 없음

```
@Controller
@RequestMapping("/board")
public class BoardController {

    @Autowired
    private boardServiceImpl boardService;

    @GetMapping("/list.do")
    public String boardList(Model model) {


        // 전체 게시물 수 구하기
        int listCount = boardService.selectListCount();
    }
}
```



게시판 구현하기 – MVC + ByBatis 연결

- BoardService

인터페이스



```
public interface boardService {  
  
    // 전체 게시물 수 구하기  
    int selectListCount();  
  
}
```

게시판 구현하기 - MVC + ByBatis 연결

• BoardServiceImpl

Service 어노테이션을 사용하여 이 클래스는 Service 목적으로 사용 되는것임을 선언

```
@Service
public class boardServiceImpl implements boardService{

    @Autowired
    private SqlSessionTemplate sqlSession;

    @Autowired
    private BoardDao boardDao;

    @Override
    public int selectListCount() {
        return boardDao.selectListCount(sqlSession);
    }
}
```

MyBatis 연결을 위한 Session과 Dao 객체 주입(생성)
* SqlSessionTemplate은 Mybatis에서 사용되는 내장 템플릿

게시판 구현하기 – MVC + ByBatis 연결

• BoardDao

Repository 어노테이션을 사용하여 이 클래스는 DAO 목적으로 사용 되는것임을 선언

```
@Repository
public class BoardDao {

    public int selectListCount(SqlSessionTemplate sqlSession) {
        return sqlSession.selectOne("boardMapper.selectListCount");
    }
}
```

네임스페이스 : boardMapper
id : selectListCount

위의 정보를 찾아 쿼리를 실행한 후 selectOne (한개의 레코드를 지정한 타입으로 반환)

_int로 선언 시 정수형 타입으로 1개만 반환됨

게시판 구현하기 – MVC + ByBatis 연결

- src/main/resources/board/board-mapper.xml

해당 mapper file에서 사용할 namespace를 선언

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTO Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="boardMapper">
  <select id="selectListCount" resultType="_int">
    SELECT COUNT(*) FROM board
  </select>
</mapper>
```

실행할 쿼리에 대한 아이디값

반환받을 값을 int 타입으로 선언

게시판 구현하기 – MVC + ByBatis 연결

- int vs _int

resultType에서는 int으로 사용하는 방식과 앞에 언더바(_)를 넣어 사용하는 방식이 있음
int의 경우 반환 타입이 Integer로 되며 _int의 경우 기본형 int로 반환됨

게시판 구현하기 – MVC + ByBatis 연결

- src/main/resources/mybatis-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE configuration PUBLIC "-//mybatis.org//DTD Config 3.0//EN" "http://mybatis.org/dtd/mybatis-3-config.dtd" >
<configuration>
  <settings>
    <setting name="jdbcTypeForNull" value="NULL" />
    <setting name="mapUnderscoreToCamelCase" value="true" />
  </settings>

  <typeAliases>
    <typeAlias type="kr.co.greenart.board.model.dto.Board" alias="board" />
  </typeAliases>

  <mappers>
    <mapper resource="/mappers/board/board-mapper.xml" />
  </mappers>
</configuration>
```

슬라이드 29p에서 확인했던 네임스페이스에 대한 별칭 설정
이 설정을 안할 경우 mapper 파일에 패키지명까지 풀네임으로 작성해야 함

사용될 mapper 파일을 정의하는 공간

게시판 구현하기 – MVC + ByBatis 연결

• DB

아래 쿼리를 실행하여 테이블, 시퀀스를 생성하고 더미 데이터를 생성

```
CREATE TABLE board (  
  b_idx NUMBER(10) PRIMARY KEY,  
  b_title VARCHAR2(50) NOT NULL,  
  b_content CLOB NOT NULL,  
  b_writer VARCHAR2(50) NOT NULL,  
  b_indate DATE DEFAULT SYSDATE,  
  b_count NUMBER NOT NULL  
);
```

```
CREATE SEQUENCE seq_b_count  
  START WITH 1  
  INCREMENT BY 1  
  NOMAXVALUE  
  NOCACHE;
```

```
CREATE SEQUENCE seq_b_idx  
  START WITH 1  
  INCREMENT BY 1  
  NOMAXVALUE  
  NOCACHE;
```

```
INSERT INTO board (b_idx, b_title, b_content, b_writer, b_count)  
VALUES (seq_b_idx.NEXTVAL, '제목2', '내용2', '김재섭', seq_b_count.NEXTVAL);
```


게시판 구현하기 – MVC + ByBatis 연결

- 결과

컨트롤러에서 listCount를 print로 출력하여 데이터 개수만큼 제대로 가져오는지 확인

게시판 구현하기 (목록 불러오기)

게시판 구현하기 - 목록 불러오기

- 추가된 자바 코드

BoardController

```
// 목록 불러오기
List<Board> list = boardService.selectListAll();

model.addAttribute("list", list); // 객체 바인딩
```

BoardDao

```
public List<Board> selectListAll(SqlSessionTemplate sqlSession) {
    return sqlSession.selectList("boardMapper.selectListAll");
}
```

BoardService

```
// 목록 불러오기
List<Board> selectListAll();
```

BoardServiceImpl

```
@Override
public List<Board> selectListAll() {
    return boardDao.selectListAll(sqlSession);
}
```

게시판 구현하기 - 목록 불러오기

• BoardService

다른곳에서 호출할 때 사용할 resultMap의 아이디

Board(DTO) 타입 사용하겠다 선언

```
<resultMap type="Board" id="boardResultSet">
  <result column="B_IDX" property="idx"></result>
  <result column="B_TITLE" property="title"></result>
  <result column="B_CONTENT" property="content"></result>
  <result column="B_WRITER" property="writer"></result>
  <result column="B_INDATE" property="indate"></result>
  <result column="B_COUNT" property="count"></result>

  <result column="ROWNUM" property="pageNum"></result>
</resultMap>
```

날짜순으로 정렬

```
<select id="selectListAll" resultMap="boardResultSet">
  SELECT * FROM board
  ORDER BY B_INDATE DESC
</select>
```

boardResultSet의 아이디를 가진 resultMap을 사용하겠다.

게시판 구현하기 - 목록 불러오기

- BoardService

사용할 프로퍼티의 이름
(ex. list.idx, list.title, list.content)

DB 테이블에 있는 컬럼의 이름

```
<resultMap type="Board" id="boardResultSet">
  <result column="B_IDX" property="idx"></result>
  <result column="B_TITLE" property="title"></result>
  <result column="B_CONTENT" property="content"></result>
  <result column="B_WRITER" property="writer"></result>
  <result column="B_INDATE" property="indate"></result>
  <result column="B_COUNT" property="count"></result>
</resultMap>

<select id="selectListAll" resultMap="boardResultSet">
  SELECT * FROM board
</select>
```

게시판 구현하기 - 목록 불러오기

- 결과

DB에서 가져온 리스트를 정상적으로 출력하는지 확인

Fixed navbar

홈게시판Disabled

Search

Search

게시판

글쓰기

번호	제목	작성자	작성일	조회수
1	제목1	김재섭	2023-06-21 16:39:27.0	1
2	제목2	김재섭	2023-06-21 16:39:40.0	2

«1»

© 2023 Company, Inc

홈게시판

게시판 구현하기 - 목록 불러오기

• 결과

년월일까지만 출력될 수 있도록 문자열 자르기

```
for(Board item : list) {  
    item.setIndate(item.getIndate().substring(0,10));  
}
```

Fixed navbar

홈

게시판

Disabled

Search

Search

게시판

글쓰기

번호	제목	작성자	작성일	조회수
1	제목1	김재섭	2023-06-21	1
2	제목2	김재섭	2023-06-21	2

«

1

»

게시판 구현하기 (페이징 처리)

게시판 구현하기 - 페이징 처리

- 더미 데이터 생성

아래 쿼리를 입력하여 총 40개의 데이터를 생성

```
INSERT INTO board (b_idx, b_title, b_content, b_writer, b_count)
SELECT seq_b_idx.NEXTVAL, '제목' || LEVEL, '내용' || LEVEL, '김재섭', seq_b_count.NEXTVAL
FROM DUAL
CONNECT BY LEVEL <= 40;
```

게시판 구현하기 - 페이징 처리

• BoardController

쿼리스트링을 전달받는 어노테이션

```
@GetMapping("/list.do")  
public String boardList(@RequestParam(value="cpage", defaultValue="1") int currentPage, Model model) {
```

// 전체 게시글 수 구하기 쿼리스트링의 이름

```
int listCount = boardService.selectListCount();
```

기본값

보여질 페이지 수 ← int pageLimit = 10;

한 페이지에 들어갈 게시글 수 ← int boardLimit = 15;

```
PageInfo pi = Pagination.getPageInfo(listCount, currentPage, pageLimit, boardLimit);  
// 페이징 처리 끝
```

```
model.addAttribute("list", list); // 객체 바인딩
```

```
model.addAttribute("pi", pi);
```

```
return "board/boardList"; // /WEB-INF/views/boardList.jsp 파일로 포워딩
```

게시판 구현하기 - 페이징 처리

- **kr.co.greenart.common.model.dto.Board**

listCount : 전체 게시글 수
currentPage : 현재 페이지
pageLimit : 한 페이지에 보여질 페이지의 수
boardLimit : 한 페이지에 들어갈 게시글의 수
maxPage : 전체 페이지
startPage : 시작 페이지
endPage : 끝 페이지

```
@Getter  
@Setter  
@NoArgsConstructor  
@AllArgsConstructor  
public class PageInfo {  
    private int listCount;  
    private int currentPage;  
    private int pageLimit;  
    private int boardLimit;  
  
    private int maxPage;  
    private int startPage;  
    private int endPage;  
}
```

게시판 구현하기 - 페이징 처리

- `kr.co.greenart.common.template.Pagination`

maxPage : 전체 페이지 수 계산

→ 전체 게시글 개수(listCount)를 한 페이지에 들어갈 게시글 수(boardLimit)로 나눈 뒤 반올림

```
public class Pagination {  
    public static PageInfo getPageInfo(int listCount, int currentPage, int pageLimit, int boardLimit) {  
        int maxPage = (int)(Math.ceil(((double)listCount/boardLimit)));  
        int startPage = (currentPage-1) / pageLimit * pageLimit+1;  
        int endPage = startPage+pageLimit-1;  
  
        if(endPage>maxPage) {  
            endPage=maxPage;  
        }  
  
        return new PageInfo(listCount, currentPage, pageLimit, boardLimit, maxPage, startPage, endPage);  
    }  
}
```

게시판 구현하기 - 페이징 처리

- **kr.co.greenart.common.template.Pagination**

startPage : 현재 페이지가 속한 범위의 시작 페이지 계산

currentPage = 16, pageLimit = 10 이라는 가정하에 계산 과정

1. $(currentPage - 1)$
→ $16 - 1 = 15$
2. $(currentPage - 1) / pageLimit$
→ $15 / 10 = 1(\text{몫})$
3. $(currentPage - 1) / pageLimit * pageLimit$
→ $1 * 10 = 10$
4. $(currentPage - 1) / pageLimit * pageLimit + 1$
→ $10 + 1 = 11$

게시판 구현하기 – 페이징 처리

- **kr.co.greenart.common.template.Pagination**

endPage : 현재 페이지가 속한 범위의 끝 페이지 계산

ex) 시작 페이지가 15며 boardLimit이 10일 경우
 $11 + 10 - 1 : 20$

게시판 구현하기 - 페이징 처리

- BoardController

인자값 pi 작성

```
// 목록 불러오기  
List<Board> list = boardService.selectListAll(pi);
```

- BoardService

매개변수 및 인자값 pi 작성

```
@Override  
public List<Board> selectListAll(PageInfo pi) {  
    return boardDao.selectListAll(sqlSession, pi);  
}
```

게시판 구현하기 - 페이징 처리

• BoardController

페이지에 맞게 가져오기 위한 offset과 limit 작성

offset : 이전에 표시한 게시글들을 건너뛰고 페이지에 맞는 게시글을 불러오기 위한 변수
RowBounds : Spring MVC에서 지원해주는 내장 클래스로써 별도의 페이징에 대한 쿼리를 작성하지 않아도 알아서 쿼리를 생성해줌

```
public List<Board> selectListAll(SqlSessionTemplate sqlSession, PageInfo pi) {  
    int offset = (pi.getCurrentPage()-1)*pi.getBoardLimit();  
  
    RowBounds rowBounds = new RowBounds(offset, pi.getBoardLimit());  
  
    return sqlSession.selectList("boardMapper.selectListAll", null, rowBounds);  
}
```


간단한 게시판 구현하기
(페이징 처리 - 버튼)

게시판 구현하기 - 버튼

• 버튼(<<)

현재 페이지가 1일 경우 버튼을 눌러도
아무런 동작을 수행하지 않도록 하는 부분


```
<nav aria-label="Page navigation example">
  <ul class="pagination justify-content-center">
    <c:choose>
      <c:when test="${pi.currentPage eq 1}">
        <li class="page-item">
          <a class="page-link" href="#" aria-label="Previous">
            <span aria-hidden="true">&laquo;</span>
          </a>
        </li>
      </c:when>
      <c:otherwise>
        <li class="page-item">
          <a class="page-link" href="list.do?cpage=${pi.currentPage-1}" aria-label="Previous">
            <span aria-hidden="true">&laquo;</span>
          </a>
        </li>
      </c:otherwise>
    </c:choose>
```

버튼 클릭 시 list.do로 요청을 하고
currentPage에 1을 뺀 값을 컨트롤러에게 전달하여
다음 페이지를 가져옴

게시판 구현하기 - 버튼

- 버튼(숫자)

시작 페이지와 끝 페이지만큼 반복



```
<c:forEach var="p" begin="${pi.startPage}" end="${pi.endPage}">
  <li class="page-item">
    <a class="page-link" href="list.do?cpage=${p}">${p}</a>
  </li>
</c:forEach>
```



반복된 페이지 번호를 생성

게시판 구현하기 - 버튼

• 버튼(<<)

마지막 페이지일 경우 버튼을 눌러도
아무런 동작을 수행하지 않도록하는 부분

```
<c:choose>
  <c:when test="${pi.currentPage eq pi.maxPage}">
    <li class="page-item">
      <a class="page-link" href="#" aria-label="Next">
        <span aria-hidden="true">&raquo;</span>
      </a>
    </li>
  </c:when>
  <c:otherwise>
    <li class="page-item">
      <a class="page-link" href="list.do?cpage=${pi.currentPage+1}" aria-label="Next">
        <span aria-hidden="true">&raquo;</span>
      </a>
    </li>
  </c:otherwise>
</c:choose>
</ul>
</nav>
```

버튼 클릭 시 list.do로 요청을 하고
currentPage에 1을 더한 값을 컨트롤러에게 전달하여
다음 페이지를 가져옴

게시판 구현하기
(페이징 처리 - 게시글 번호)

게시판 구현하기 - 게시물 번호

• BoardController

아래와 같이 계산하여 글의 번호를 뒤에서부터 보여줄 수 있도록 하기 위한 코드

전체 게시물 수 - (현재 페이지 - 1) * 한 페이지에 보여줄 게시물 수

```
@GetMapping("/list.do")
public String boardList(@RequestParam(value="cpage", defaultValue="1") int currentPage, ModelMap model) {

    // 전체 게시물 수 구하기
    int listCount = boardService.selectListCount();
    int pageLimit = 10;
    int boardLimit = 10;
    int row = listCount - (currentPage - 1) * boardLimit;

    PageInfo pi = Pagination.getPageInfo(listCount, currentPage, pageLimit, boardLimit);

    // 목록 불러오기
    List<Board> list = boardService.selectListAll(pi);

    for(Board item : list) {
        item.setIndate(item.getIndate().substring(0,10));
    }

    model.addAttribute("list", list); // 객체 바인딩
    model.addAttribute("row", row);
    model.addAttribute("pi", pi);

    return "board/boardList"; // /WEB-INF/views/boardList.jsp 파일로 포워딩
}
```

게시판 구현하기 - 게시물 번호

- BoardList.jsp

한번 반복될 때 마다 row의 값을 1씩 줄임

```
<c:otherwise>
  <c:forEach var="item" items="${list}">
    <tr>
      <td>${row}</td>
      <td>${item.title}</td>
      <td>${item.writer}</td>
      <td>${item.indate}</td>
      <td>${item.count}</td>
    </tr>
    <c:set var="row" value="${row - 1}" />
  </c:forEach>
</c:otherwise>
</c:choose>
```