

5장. 회원 및 게시판 관리 시스템



Member And Board



Spring Boot

회원 및 게시판 관리 시스템

Good Releation - 좋은 사이



[로그인](#) | [회원 가입](#) | [회원 목록](#) | [게시판](#)



회원 및 게시판 관리 시스템

▪ index.html

```
<section id="container">
  <div id="main">
    <h2>Good Releation - 좋은 사이</h2>
    <div class="main-pic">
      
    </div>
    <div class="menu">
      <th:block th:if="${session.loginEmail}">
        <span style="color: ■ blue"
          th:text="'[' + ${session.loginName} + '님]'"></span>
        <a th:href="@{/members/logout}" th:text='로그아웃'></a> |
      </th:block>
      <th:block th:unless="${session.loginEmail}">
        <a th:href="@{/members/login}" th:text='로그인'></a> |
      </th:block>

      <a th:href="@{/members/join}">회원 가입</a> |
      <a th:href="@{/members}">회원 목록</a> |
      <a th:href="@{/boards/pages}">게시판</a>
    </div>
  </div>
</section>
```



회원 및 게시판 관리 시스템

글쓰기

가입 인사

안녕하세요~
잘 부탁드립니다.

김기용

등록

취소



회원 및 게시판 관리 시스템

▪ BoardDTO

```
@Data
public class BoardDTO {
    private Long id; //글번호

    @NotEmpty(message="제목은 필수 항목입니다.")
    private String title; //글제목

    @NotEmpty(message="내용은 필수 항목입니다.")
    private String content; //글내용

    private String writer; //글쓴이

    private Integer hits; //조회수

    private Timestamp regDate; //작성일
}
```



회원 및 게시판 관리 시스템

▪ Board

```
@Data
@Table(name = "t_board")
@Entity
public class Board {
    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    private Long id;          //글번호

    @Column(nullable=false)
    private String title;     //글제목

    @Column(length=2000, nullable=false)
    private String content;   //글내용

    @Column(length=30, nullable=false)
    private String writer;    //글쓴이
```



회원 및 게시판 관리 시스템

▪ Board

```
@Column(columnDefinition = "Integer default 0")
private Integer hits;           //조회수

@CreationTimestamp
private Timestamp regDate; //작성일

//Entity에 DTO 저장 메서드
public static Board toSaveEntity(BoardDTO dto) {
    Board board = new Board();
    board.setTitle(dto.getTitle());
    board.setContent(dto.getContent());
    board.setWriter(dto.getWriter());
    board.setHits(0);

    return board;
}
```



회원 및 게시판 관리 시스템

▪ BoardRepository

```
public interface BoardRepository extends JpaRepository<Board, Long>{  
  
    //조회수 증가 쿼리 메서드  
    @Modifying //삽입, 수정등의 변경이 있을때 사용  
    @Query(value="update Board b set b.hits=b.hits+1 where b.id=:id")  
    void updateHits(Long id);  
}
```



회원 및 게시판 관리 시스템

▪ BoardService

```
@RequiredArgsConstructor
@Service
public class BoardService {
    private final BoardRepository repository;

    //글쓰기
    public void save(BoardDTO dto) {
        Board board = Board.toSaveEntity(dto);
        repository.save(board);
    }

    //글 목록
    public List<Board> findAll() {
        //오름차순 정렬: findAll()
        //내림차순 정렬:
        Sort order = Sort.by(Sort.Direction.DISC, "id");
        return repository.findAll(order);
    }
}
```



회원 및 게시판 관리 시스템

■ BoardService

```
//글 목록(페이지 처리)
public Page<Board> findAll(Pageable pageable) {
    //pageable = PageRequest.of(0, 10); //오름차순
    int page = pageable.getPageNumber() - 1;
    int pageSize = 10;

    Log.info("--> pageable.getPageNumber(): " + pageable.getPageNumber());
    //브라우저 확인 - /boards/pages?page=3
    pageable = PageRequest.of(page, pageSize, Sort.Direction.DISC, "id");

    Page<Board> boardList = repository.findAll(pageable);
    //getNumber는 pageNumber 보다 1작음
    Log.info("--> boardList.getNumber(): " + boardList.getNumber());

    return boardList;
}

//글 상세 보기
public Board findById(Long id) {
    return repository.findById(id)
        .orElseThrow(() ->
            new IllegalArgumentException("해당 글이 존재하지 않습니다. ID=" + id));
}
```



회원 및 게시판 관리 시스템

■ BoardService

```
//글 삭제
public void delete(Long id) {
    repository.deleteById(id);
}

//글 수정
public void update(BoardDTO dto) {
    //1. 수정할 게시글 가져옴(수정 폼에서 id를 hidden으로 넘겨받음)
    Board board = repository.findById(dto.getId())
        .orElseThrow(() ->
            new IllegalArgumentException("해당 글이 존재하지 않습니다."));

    //2. 제목, 내용 수정
    board.setTitle(dto.getTitle());
    board.setContent(dto.getContent());
    repository.save(board);
}

@Transactional //트랜잭션(조회수, 상세보기) 완료
public void updateHits(Long id) {
    repository.updateHits(id);
}
```



회원 및 게시판 관리 시스템

▪ BoardController

```
@Slf4j
@RequiredArgsConstructor
@RequestMapping("/boards")
@Controller
public class BoardController {
    private final BoardService service;

    //글쓰기 페이지
    @GetMapping("/write")
    public String writeForm(BoardDTO boardDTO) {
        Log.info("Has errors...");
        return "board/write";
    }

    //글쓰기 처리
    @PostMapping("/write")
    public String write(@Valid BoardDTO dto,
        BindingResult bindingResult) {
        if(bindingResult.hasErrors()) {
            return "board/write";
        }
        service.save(dto);
        return "redirect:/boards";
    }
}
```



회원 및 게시판 관리 시스템

▪ BoardController

```
//글 목록
@GetMapping
public String getBordList(Model model){
    List<Board> boardList = service.findAll();
    model.addAttribute("boardList", boardList);
    return "board/list";
}

//글목록(페이지 처리)
//boards/pages?page=0
@GetMapping("/pages")
public String getBoardPage(Model model,
    @PageableDefault(page=1) Pageable pageable) {

    Page<Board> boardList = service.findAll(pageable);

    //하단의 페이지 블럭
    int blockLimit = 10;
```



회원 및 게시판 관리 시스템

■ BoardController

```
//시작 페이지 - 1, 2, 3...
//시작 번호 - 1, 11, 21...
//예) 번호 - 13, 13/10-1.3 -> 2(올림)-1 * 10 + 1 => 11
int startPage
    = ((int)Math.ceil((double)pageable.getPageNumber() / blockLimit)-1)
      * blockLimit + 1;
//페이지의 행번호(끝) 10, 20, 30...
//int endPage = startPage + blockLimit - 1;
int endPage = (startPage + blockLimit - 1) > boardList.getTotalPages() ?
    boardList.getTotalPages() : (startPage + blockLimit - 1);

model.addAttribute("boardList", boardList);
model.addAttribute("startPage", startPage);
model.addAttribute("endPage", endPage);

return "board/pages";
}
```



회원 및 게시판 관리 시스템

▪ BoardController

```
//글 상세보기
@GetMapping("/{id}")
public String getBoard(@PathVariable Long id,
    @PageableDefault(page=1) Pageable pageable,
    Model model) {
    try {
        service.updateHits(id); //조회수 증가

        Board board = service.findById(id); //글 상세 보기
        model.addAttribute("board", board);
        model.addAttribute("page", pageable.getPageNumber());
        return "board/detail";
    } catch (Exception e) {
        model.addAttribute("errorMsg", e.getMessage());
        return "error/errorPage";
    }
}
```



회원 및 게시판 관리 시스템

▪ BoardController

```
//글 삭제
@GetMapping("/delete/{id}")
public String deleteBoard(@PathVariable Long id) {
    service.delete(id);
    return "redirect:/boards/pages";
}

//글 수정 페이지
@GetMapping("/update/{id}")
public String updateForm(@PathVariable Long id,
    Model model) {
    Board board = service.findById(id);
    model.addAttribute("board", board);
    return "board/update";
}

//글 수정 처리
@PostMapping("/update")
public String update(@ModelAttribute BoardDTO dto) {
    service.update(dto);
    return "redirect:/boards/" + dto.getId();
}
```



회원 및 게시판 관리 시스템

- 글쓰기(세션이 없는 경우)

localhost:8080의 메시지

로그인이 필요합니다.

확인

로그인

이메일

비밀번호

로그인



회원 및 게시판 관리 시스템

▪ style.css

```
/* boardList 스타일 */
.btnList{margin: 16px;}
.btnList, button{padding: 5px 10px;}
.pagination{
    margin: 10px;
    text-align: center;
}
.pagination a{
    padding: 4px;
    text-decoration: none;
}

/* write 스타일 */
.btnWrite{
    width: 500px;
    margin: 0 auto;
    text-align: right;
}
.btnWrite button{padding: 8px 12px;}
```

```
.writeForm, .boardDetail, .boardUpdate{
    width: 500px;
    margin: 0 auto;
}
.writeForm ul, .boardDetail ul,
.boardUpdate ul{padding-left: 0;}
.writeForm ul li{margin: 0;}

.writeForm textarea, .boardDetail textarea,
.boardUpdate textarea{
    resize: none;
    margin: 0;
}
```



회원 및 게시판 관리 시스템

▪ write.html

```
<script th:inline="javascript">
    // 세션이 없으면 로그인 요청
    let loginEmail = [[${session.loginEmail}]];
    if (!loginEmail) {
        alert("로그인이 필요합니다.");
        location.href = "/members/login";
    }
</script>

<section id="container">
    <h2>글쓰기</h2>
    <!-- validation -->
    <form th:action="@{/boards/write}" method="post" class="writeForm"
          th:object="${boardDTO}">
        <div th:if="${#fields.hasAnyErrors()}">
            <div th:each="err : ${#fields.allErrors()}"
                  th:text="${err}" class="error"></div>
        </div>
```



회원 및 게시판 관리 시스템

▪ write.html

```
<fieldset>
  <ul>
    <li>
      <input type="text" th:field="*{title}" size=62
        placeholder="제목">
    </li>
    <li>
      <textarea rows="8" cols="60" th:field="*{content}"
        placeholder="내용"></textarea>
    </li>
    <li>
      <input type="text" name="writer" size=62
        th:value="${session.loginName}">
    </li>
    <li>
      <input type="submit" value="등록">
      <input type="reset" value="취소">
    </li>
  </ul>
</fieldset>
</form>
</section>
```



회원 및 게시판 관리 시스템

게시판(페이징)

번호	제목	글쓴이	조회수	작성일
2	가을 옷 준비	박선화	0	2025-10-24 05:36
1	가입 인사	김기용	0	2025-10-24 04:48

[Prev](#) [1](#) [Next](#)

글쓰기



회원 및 게시판 관리 시스템

▪ list.html

```
<h2>게시판</h2>
<table class="boardList">
  <thead>
    <tr>
      <th>번호</th><th>제목</th><th>글쓴이</th><th>조회수</th><th>작성일</th>
    </tr>
  </thead>
  <tbody>
    <tr th:each="b: ${boardList}">
      <td th:text="${b.id}"></td>
      <td>
        <a th:href="@{/boards/{id}(id=${b.id})}" th:text="${b.title}"></a>
      </td>
      <td th:text="${b.writer}"></td>
      <td th:text="${b.hits}"></td>
      <td th:text="${#dates.format(b.regDate, 'yyyy-MM-dd HH:mm')}"></td>
    </tr>
  </tbody>
</table>
<div class="btnWrite">
  <a th:href="@{/boards/write}"><button>글쓰기</button></a>
</div>
```



회원 및 게시판 관리 시스템

- 세션이 있는 경우

글 상세 보기

가을 옷 준비

날씨가 쌀쌀합니다.
가을 옷을 준비해요...

글쓴이: 박선화(작성일: 2025-10-24 05:36:11.551231)

조회수: 1

수정

삭제

목록



회원 및 게시판 관리 시스템

- 세션이 없는 경우

글 상세 보기

가입 인사

안녕하세요~
반갑습니다..

글쓴이: 김기용(작성일: 2025-10-24 04:48:31.294297)

조회수: 1

목록



회원 및 게시판 관리 시스템

▪ detail.html

```
<h2>글 상세 보기</h2>
<div class="boardDetail">
  <fieldset>
    <ul>
      <li>
        <input type="text" name="title" size=62
          th:value="${board.title}" readonly>
      </li>
      <li>
        <textarea rows="8" cols="60" name="content"
          readonly th:text="${board.content}"></textarea>
      </li>
      <li>
        <input type="text" name="writer" size=62 readonly
          th:value="'글쓴이: ' + ${board.writer} + '(작성일: '
            + ${board.regDate} + ')">
      </li>
    </ul>
  </fieldset>
</div>
```



회원 및 게시판 관리 시스템

▪ detail.html

```
<li>
    <input type="text" name="writer" size=62 readonly
           th:value="' 조회수: ' + ${board.hits}">
</li>
<li>
    <th:block th:if="${session.loginName == board.writer}">
        <a th:href="@{/boards/update/${board.id}|"><button>수정</button></a>
        <a th:href="@{/boards/delete/${board.id}|"
           onclick="return confirm('정말 삭제하시겠습니까?')"><button>삭제</button></a>
    </th:block>
    <a th:href="@{/boards/pages}"><button>목록</button></a>
</li>
</ul>
```



회원 및 게시판 관리 시스템

게시글 수정

가을 옷 준비 - 수정

날씨가 쌀쌀합니다.
가을 옷을 준비해요..
수정합니다.

글쓴이: 박선화(작성일: 2025-10-24 05:36:11.551231)

조회수: 1

저장

취소



회원 및 게시판 관리 시스템

▪ update.html

```
<section id="container">
  <h2>게시글 수정</h2>
  <form th:action="@{/boards/update}" method="post" class="boardUpdate">
    <input type="hidden" name="id" th:value="${board.id}">
    <fieldset>
      <ul>
        <li>
          <input type="text" name="title" size=62
            th:value="${board.title}">
        </li>
        <li>
          <textarea rows="8" cols="60" name="content"
            th:text="${board.content}"></textarea>
        </li>
        <li>
          <input type="text" name="writer" size=62 readonly
            th:value="'글쓴이: ' + ${board.writer} + '(작성일: '
              + ${board.regDate} + ')">
        </li>
      </ul>
    </fieldset>
  </form>
```



회원 및 게시판 관리 시스템

▪ update.html

```
<li>
    <input type="text" name="writer" size=62 readonly
           th:value="' 조회수: ' + ${board.hits}">
</li>
<li>
    <button type="submit">저장</button>
    <button type="reset">취소</button>
</li>
</ul>
</fieldset>
</form>
```



회원 및 게시판 관리 시스템

게시판(페이징)

번호	제목	글쓴이	조회수	작성일
123	테스트 제목 123	tester	0	2025-10-24 06:14
122	테스트 제목 122	tester	0	2025-10-24 06:14
121	테스트 제목 121	tester	0	2025-10-24 06:14
120	테스트 제목 120	tester	0	2025-10-24 06:14
119	테스트 제목 119	tester	0	2025-10-24 06:14
118	테스트 제목 118	tester	0	2025-10-24 06:14
117	테스트 제목 117	tester	0	2025-10-24 06:14
116	테스트 제목 116	tester	0	2025-10-24 06:14
115	테스트 제목 115	tester	0	2025-10-24 06:14
114	테스트 제목 114	tester	0	2025-10-24 06:14

Prev 1 2 3 4 5 6 7 8 9 10 Next

글쓰기



회원 및 게시판 관리 시스템

■ 테스트 – 페이지 처리

```
@SpringBootTest
public class BoardServiceTest {

    @Autowired
    private BoardService service;

    @Test
    void testInsertBoard() {
        for(int i=1; i<=123; i++) {
            BoardDTO dto = new BoardDTO();
            dto.setTitle("테스트 제목 " + i);
            dto.setContent("테스트 내용 ");
            dto.setWriter("tester");

            service.save(dto);
        }
    }
}
```



회원 및 게시판 관리 시스템

■ 테스트 – 페이지 처리

hits	id	reg_date	writer	content	title
0	123	2025-10-24 06:14:24.711873	tester	테스트 내용	테스트 제목 123
0	122	2025-10-24 06:14:24.705134	tester	테스트 내용	테스트 제목 122
0	121	2025-10-24 06:14:24.698230	tester	테스트 내용	테스트 제목 121
0	120	2025-10-24 06:14:24.690780	tester	테스트 내용	테스트 제목 120
0	119	2025-10-24 06:14:24.683413	tester	테스트 내용	테스트 제목 119
0	118	2025-10-24 06:14:24.675870	tester	테스트 내용	테스트 제목 118
0	117	2025-10-24 06:14:24.670041	tester	테스트 내용	테스트 제목 117
0	116	2025-10-24 06:14:24.662454	tester	테스트 내용	테스트 제목 116
0	115	2025-10-24 06:14:24.655743	tester	테스트 내용	테스트 제목 115
0	114	2025-10-24 06:14:24.649551	tester	테스트 내용	테스트 제목 114
0	113	2025-10-24 06:14:24.642543	tester	테스트 내용	테스트 제목 113
0	112	2025-10-24 06:14:24.634805	tester	테스트 내용	테스트 제목 112
0	111	2025-10-24 06:14:24.627812	tester	테스트 내용	테스트 제목 111
0	110	2025-10-24 06:14:24.604088	tester	테스트 내용	테스트 제목 110
0	109	2025-10-24 06:14:24.595702	tester	테스트 내용	테스트 제목 109
0	108	2025-10-24 06:14:24.588267	tester	테스트 내용	테스트 제목 108
0	107	2025-10-24 06:14:24.581483	tester	테스트 내용	테스트 제목 107
0	106	2025-10-24 06:14:24.574388	tester	테스트 내용	테스트 제목 106
0	105	2025-10-24 06:14:24.568036	tester	테스트 내용	테스트 제목 105
0	104	2025-10-24 06:14:24.561033	tester	테스트 내용	테스트 제목 104



회원 및 게시판 관리 시스템

▪ pages.html

```
<h2>게시판(페이징)</h2>
<table class="boardList">
  <thead>
    <tr>
      <th>번호</th><th>제목</th><th>글쓴이</th><th>조회수</th><th>작성일</th>
    </tr>
  </thead>
  <tbody>
    <tr th:each="b: ${boardList}">
      <td th:text="${b.id}"></td>
      <td>
        <a th:href="@{/boards/{id}(id=${b.id})}" th:text="${b.title}"></a>
      </td>
      <td th:text="${b.writer}"></td>
      <td th:text="${b.hits}"></td>
      <td th:text="${#dates.format(b.regDate, 'yyyy-MM-dd HH:mm')}"></td>
    </tr>
  </tbody>
</table>
```



회원 및 게시판 관리 시스템

▪ pages.html

```
<!-- 페이지 영역 -->
<div class="pagination">
  <!-- 이전 페이지, 예) 2페이지 클릭, db는 1page임 -->
  <a th:href="${boardList.first} ? '#' :
    @{/boards/pages(page=${boardList.number})}" th:text="Prev"></a>
  <!-- 페이지 번호 블록
    처음: ${#numbers.sequence(1, 5)} -->
  <span th:each="page : ${#numbers.sequence(startPage, endPage)}">
    <!-- 번호를 누르면 페이지 바뀜 -->
    <a th:href="@{/boards/pages(page=${page})}"
      th:text="${page}"></a>
  </span>
  <!-- 다음 페이지 -->
  <a th:href="${boardList.last} ? '#' :
    @{/boards/pages(page=${boardList.number + 2})}" th:text="Next"></a>
</div>

<!-- 글쓰기 버튼 -->
<div class="btnWrite">
  <a th:href="@{/boards/write}"><button>글쓰기</button></a>
</div>
```



회원 및 게시판 관리 시스템

- 페이지 처리 - 글 목록, 상세보기 수정

pages.html

```
<tr th:each="b: ${boardList}">
  <td th:text="${b.id}"></td>
  <td>
    <a th:href="@{/boards/${b.id}/(page=${boardList.number+1})}"
      th:text="${b.title}"></a>
  </td>
  <td th:text="${b.writer}"></td>
```

detail.html

```
<th:block th:if="${session.loginName == board.writer}">
  <a th:href="@{/boards/update/${board.id}/}"><button>수정</button></a>
  <a th:href="@{/boards/delete/${board.id}/}"
    onclick="return confirm('정말 삭제하시겠습니까?')"><button>삭제</button></a>
</th:block>
<a th:href="@{/boards/pages(page=${page})}"><button>목록</button></a>
```



회원 및 게시판 관리 시스템

- 페이지 처리 - 상세보기 수정

BoardController

```
@GetMapping("/{id}")
public String getBoard(@PathVariable Long id,
    @PageableDefault(page=1) Pageable pageable,
    Model model) {
    try {
        //조회수 증가
        service.updateHits(id);

        //글 상세 보기
        Board board = service.findById(id);
        model.addAttribute("board", board);
        model.addAttribute("page", pageable.getPageNumber());
        return "board/detail";
    } catch (Exception e) {
        model.addAttribute("errorMsg", e.getMessage());
        return "error/404.html";
    }
}
```

모델로 저장



회원 및 게시판 관리 시스템

게시글 목록

번호	제목	글쓴이	조회수	작성일
1	가입 인사	김기용	3	2025-10-24 06:28

[Prev](#) [1](#) [Next](#)

회원 및 게시판 관리 시스템

- 제목만 검색 – pages.html

```
<h2>게시글 목록</h2>
<!-- 검색 영역 -->
<!-- 제목만 검색할 경우 -->
<form th:action="@{/boards/pages}" method="get"
      class="searchForm">
    <input type="text" name="keyword" th:value="${kw}">
    <input type="submit" value="검색">
</form>
```

- 게시글 목록 - Repository

```
//제목 검색어가 포함된 게시글 목록을 페이지 처리하여 조회 - Page<T> 사용
Page<Board> findByTitleContaining(String keyword, Pageable pageable);
```



회원 및 게시판 관리 시스템

■ 게시글 목록 - Service

```
public Page<Board> findByTitleContaining(String keyword, Pageable pageable) {  
    int page = pageable.getPageNumber() - 1;  
    int pageSize = 10;  
  
    pageable = PageRequest.of(page, pageSize, Sort.Direction.DESC, "id");  
  
    Page<Board> boardList =  
        repository.findByTitleContaining(keyword, pageable);  
  
    return boardList;  
}
```



회원 및 게시판 관리 시스템

■ 게시물 목록 - Controller

```
@GetMapping("/pages")
public String getBoardPage(
    //false -> 키워드가 없어도 목록 보임, required -> true이면 오류 발생
    @RequestParam(value="keyword", required=false) String keyword,
    @PageableDefault(page=1) Pageable pageable,
    Model model) {
    Page<Board> boardList = null;
    if(keyword == null) {
        boardList = service.findAll(pageable);
    }else {
        boardList = service.findByTitleContaining(keyword, pageable);
    }
}
```

```
model.addAttribute("boardList", boardList);
model.addAttribute("startPage", startPage);
model.addAttribute("endPage", endPage);
model.addAttribute("kw", keyword); //검색어 보냄

return "board/pages";
}
```



게시글 목록

내용 ▾

번호	제목	글쓴이	조회수	작성일
2	가을 옷 준비	우영우	4	2025-10-24 06:29

[Prev](#) [1](#) [Next](#)



게시글 검색

■ 제목과 내용 검색 – pages.html

```
<!-- 제목과 내용을 검색할 경우 -->
<form th:action="@{/boards/pages}" method="get"
      class="searchForm">
  <select name="type">
    <option th:value="title" th:text="제목"
            th:selected="${type eq 'title'}"></option>
    <option th:value="content" th:text="내용"
            th:selected="${type eq 'content'}"></option>
  </select>
  <input type="text" name="keyword" th:value="${kw}">
  <input type="submit" value="검색">
</form>
```

■ 게시글 목록 - Repository

```
//제목 및 내용 검색어가 포함된 게시글 목록을 페이지 처리하여 조회 - Page<T> 사용
Page<Board> findByContentContaining(String keyword, Pageable pageable);
```



■ 게시글 목록 - Controller

```
@GetMapping("/pages")
public String getBoardPage(
    //false -> 키워드가 없어도 목록 보임, required -> true이면 오류 발생
    @RequestParam(value="keyword", required=false) String keyword,
    @RequestParam(value="type", required=false) String type,
    @PageableDefault(page=1) Pageable pageable,
    Model model) {
    Page<Board> boardList = null;
    if(keyword == null) {
        boardList = service.findAll(pageable);
    }else if(keyword != null && type.equals("title")){
        boardList = service.findByTitleContaining(keyword, pageable);
    }else if(keyword != null && type.equals("content")) {
        boardList = service.findByContentContaining(keyword, pageable);
    }
}
```



회원 및 게시판 관리 시스템

■ 게시물 목록 - Controller

```
model.addAttribute("boardList", boardList);  
model.addAttribute("startPage", startPage);  
model.addAttribute("endPage", endPage);  
model.addAttribute("type", type); //검색 유형  
model.addAttribute("kw", keyword); //검색어  
  
return "board/pages";  
}
```



회원 및 게시판 관리 시스템

■ 게시글 목록 - Service

```
public Page<Board> findByContentContaining(String keyword, Pageable pageable) {  
    int page = pageable.getPageNumber() - 1;  
    int pageSize = 10;  
  
    pageable = PageRequest.of(page, pageSize, Sort.Direction.DESC, "id");  
  
    Page<Board> boardList =  
        repository.findByContentContaining(keyword, pageable);  
  
    return boardList;  
}
```

