게시판/홈페이지

스프링 부트

spring Boot

2조

이원진,이지환,장순재

추형욱 한형빈

Chapter.1

목차

목표수립

계획단계

역할분담

구현

LOGO

프로젝트목표

스프링부트sts4.0 Detail

- boot만의 특징을 이해한다
- CRUD가 가능한 홈페이지
- 검색/공지등 추가편의기능

Plan 계획

1차구현

List를 통한 페이징구현
CRUD기능구현
검색기능
글/댓글기능분리

테스트및정리

Junit을통한더미데이터
기능오류확인
오류수정및 분석

기본구조설계

Entity생성
Controller/Service생성
Repository추가

추가기능구현

로그인관련기능
보안처리
공지기능생성

Chapter.2

역할분담

이지환

기본설계
댓글CRUD구현
로그인기능구현

이원진

더미데이터삽입
Html페이지구현
추천기능구현

장순재

게시판CRUD구현
각종Class구현
기본페이지기능

한형빈

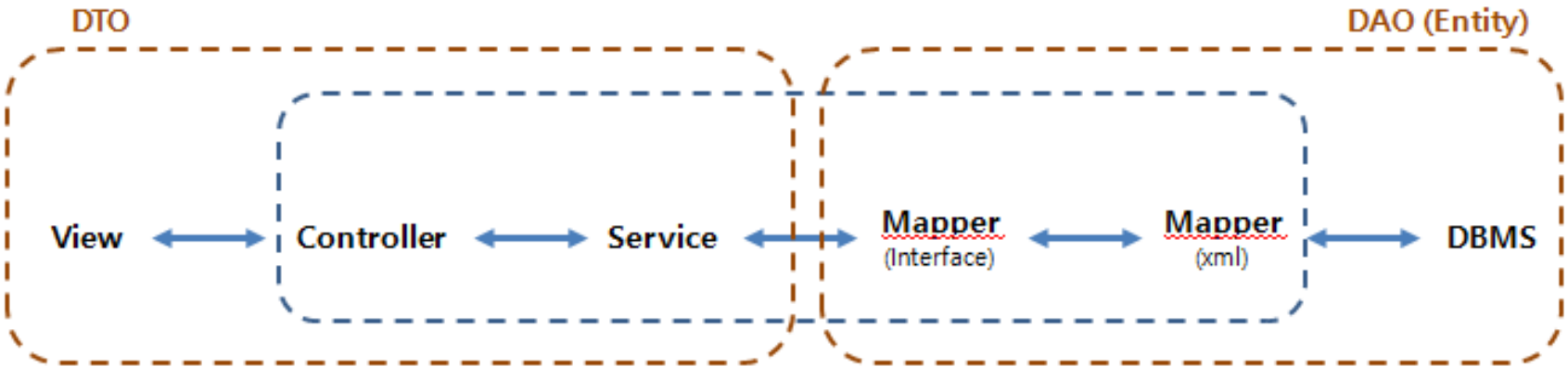
디버깅
검색기능구현
네비게이션바구현

추형욱

페이징기능개선
공지게시판기능추가
스프링시큐리티
권한관리
ppt

LOGO

설계 계획


DTO
DAO (Entity)
View
Controller
Service
Mapper
(Interface)
Mapper
(xml)
DBMS

LOGO

기본설계

설계 계획

ENTITY

01 테이블구성
02 리포지터리구현

Controller

01 데이터 처리
02 CRUD기능
03 주소맵핑등

Service

01Controller내부 기능
02 쿼리문 처리
03 오류처리

LOGO

기본설계

코드에서

```java
@Getter
@Setter
@Entity
public class Question {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer id;

    @Column(length = 200)
    private String subject;

    @Column(columnDefinition = "TEXT")
    private String content;

    private LocalDateTime createDate;

    @OneToMany(mappedBy = "question", cascade = CascadeType.REMOVE)
    private List<Answer> answerList;
```

Entity

h2CONSOLE을이용 내부 DB를이용해서 처리

일부처리는 Sql문법대신 어노테이션으로처리

H2콘솔 내부에서 sql문 도실행이가능

기본설계

코드예시

```
@Query("select "
        + "distinct q "
        + "from Question q "
        + "left outer join SiteUser u1 on q.author=u1 "
        + "left outer join Answer a on a.question=q "
        + "left outer join SiteUser u2 on a.author=u2 "
        + "where "
        + "   q.subject like %:kw% "
        + "   or q.content like %:kw% "
        + "   or u1.username like %:kw% "
        + "   or a.content like %:kw% "
        + "   or u2.username like %:kw% ")
```

```
y<Question, Integer>{
Question findBySubjectAndContent(String subject, String content);
List<Question> findBySubjectLike(String subject);
Page<Question> findAll(Pageable pageable);
Page<Question> findAll(Specification<Question> spec, Pageable pageable);
```

Repository

Service에 쓰는 기능을

처리해주는곳[인터페이스]

대부분 이곳을통해처리

필요에따라 쿼리문을
넣기도한다

기본설계

코드예시

```
private Specification<Question> search(String kw) {
        return new Specification<>() {
            private static final long serialVersionUID = 1L;
            @Override
            public Predicate toPredicate(Root<Question> q, CriteriaQuery<?> query, CriteriaBuilder cb
{
                query.distinct(true);  // 중복을 제거
                Join<Question, SiteUser> u1 = q.join("author", JoinType.LEFT);
                Join<Question, Answer> a = q.join("answerList", JoinType.LEFT);
                Join<Answer, SiteUser> u2 = a.join("author", JoinType.LEFT);
                return cb.or(cb.like(q.get("subject"), "%" + kw + "%"), // 제목
                        cb.like(q.get("content"), "%" + kw + "%"),      // 내용
                        cb.like(u1.get("username"), "%" + kw + "%"),    // 질문 작성자
                        cb.like(a.get("content"), "%" + kw + "%"),      // 답변 내용
                        cb.like(u2.get("username"), "%" + kw + "%"));   // 답변 작성자
            }
```

기본설계

코드예시

```
@PreAuthorize("isAuthenticated()")
        @GetMapping("/modify/{id}")
        public String questionModify(QuestionForm questionForm, @PathVariable("id") Integer id,
Principal principal) {
            Question question = this.questionService.getQuestion(id);
            if(!question.getAuthor().getUsername().equals(principal.getName())) {
                throw new ResponseStatusException(HttpStatus.BAD_REQUEST, "수정권한이 없습니다.");
            }
            questionForm.setSubject(question.getSubject());
            questionForm.setContent(question.getContent());
            return "question_form";
        }
    }
```

기본설계

코드예시

Html부

실질적 출력 하는곳

```
<tr class="text-center" th:each="question, loop : ${paging}">
        <td th:text="${paging.getTotalElements - (paging.number * paging.size) -  loop.index}"></td>
        <td class="text-start">
            <a th:href="@{|/q/detail/${question.id}|}" th:text="${question.subject}"></a>
            <span class="text-danger small ms-2" th:if="${#lists.size(question.answerList) > 0}"
th:text="${#lists.size(question.answerList)}"></span>
        </td>
        <td><span th:if="${question.author != null}" th:text="${question.author.username}"></span></td>
        <td th:text="${#temporals.format(question.createDate, 'yyyy-MM-dd HH:mm')}"></td>
    </tr>
</tbody>
```

KB좋아 로그인 회원가입

질문 등록하기

찾기

번호	제목	글쓴이	작성일시
[공지]	4차공지	admin	2024-05-22 15:53
[공지]	3번째공지	admin	2024-05-22 12:16
[공지]	2차공지작성	admin	2024-05-22 12:02
208	테스트 데이터입니다:[100]		2024-05-23 09:13
207	테스트 데이터입니다:[099]		2024-05-23 09:13
206	테스트 데이터입니다:[098]		2024-05-23 09:13
205	테스트 데이터입니다:[097]		2024-05-23 09:13
204	테스트 데이터입니다:[096]		2024-05-23 09:13
203	테스트 데이터입니다:[095]		2024-05-23 09:13
202	테스트 데이터입니다:[094]		2024-05-23 09:13
201	테스트 데이터입니다:[093]		2024-05-23 09:13
200	테스트 데이터입니다:[092]		2024-05-23 09:13
199	테스트 데이터입니다:[091]		2024-05-23 09:13

이것은 테스트용입니다

작성중 작성2 작성3 테스트4

modified at
2024-05-23 11:02

testtest1122
2024-05-21 15:48

추천 0 수정 삭제

0개의 답변이 있습니다.

답변등록

추가기능

설계 계획

게시판기능

01 list/paging
02 검색등부가기능
03 공지작성기능

스프링시큐리티

01 기능암호화
02 접근권한설정

로그인기능

01아이디생성/삭제
02 로그인/로그아웃
03 권한[CRUD]

LOGO

페이징

코드예시

```
public Page<Question> getList(int page, String kw) {
        List<Sort.Order> sorts = new ArrayList<>();
        sorts.add(Sort.Order.desc("createDate"));
        Pageable pageable = PageRequest.of(page, 10, Sort.by(sorts));
        return this.questionRepository.findAllByKeyword(kw, pageable);
    }
```

페이징처리

실질적 출력 하는html과

Service,repository등
에서 처리하는부분존재

Service에 Getlist 생성
Page처리를 한다

Controller부분도 수정

```
@GetMapping("/list")
  public String list(Model model, @RequestParam(value="page", defaultValue="0") int page) {
      Page<Question> paging = this.questionService.getList(page);
      model.addAttribute("paging", paging);
      return "question_list";
```

```
<div th:if="${!paging.isEmpty()}">
    <ul class="pagination justify-content-center">
        <li class="page-item" th:classappend="${!paging.hasPrevious} ? 'disabled'">
            <a class="page-link" href="javascript:void(0)" th:data-page="${paging.number > 9 ?
paging.number-10 : 0}">
                <span>이전</span>
            </a>
        </li>
        <li th:each="page: ${#numbers.sequence(paging.number/10 * 10, (paging.number/10 * 10 + 9) <
paging.totalPages ? (paging.number/10 * 10 + 9) : paging.totalPages - 1)}"
            th:classappend="${page == paging.number} ? 'active'" class="page-item">
            <a th:text="${page+1}" class="page-link" href="#" th:onclick="'javascript:void(0);'"
th:data-page="${page}"></a>
        </li>
        <li class="page-item" th:classappend="${!paging.hasNext} ? 'disabled'">
            <a class="page-link" href="javascript:void(0)" th:data-page="${paging.number+10 >
paging.totalPages? paging.totalPages - 1 :paging.number+10} ">
                <span>다음</span>
            </a>
        </li>
    </ul>
</div>
```

페이징

코드예시

```
        <li th:each="page: ${#numbers.sequence(paging.number/10 * 10, (paging.number/10 * 10 + 9) <
paging.totalPages ? (paging.number/10 * 10 + 9) : paging.totalPages - 1)}"
            th:classappend="${page == paging.number} ? 'active'" class="page-item">
            <a th:text="${page+1}" class="page-link" href="#" th:onclick="'javascript:void(0);'"
th:data-page="${page}"></a>
        </li>
```

```
        <li class="page-item" th:classappend="${!paging.hasNext} ? 'disabled'">
            <a class="page-link" href="javascript:void(0)" th:data-page="${paging.number+10 >
 paging.totalPages? paging.totalPages - 1 :paging.number+10} ">
                <span>다음</span>
            </a>
        </li>
```

페이징

코드예시

```
<ul class="nav nav-tabs">
    <li class="nav-item" onclick="openCategory(event)">
        <a class="nav-link"  aria-current="page"  href="/">전체</a>
    </li>
    <li class="nav-item" onclick="openCategory(event)">
        <a class="nav-link"  href="/board/category/notification">공지</a>
    </li>
    <li class="nav-item" onclick="openCategory(event)">
        <a class="nav-link"  href="/board/category/joboffer">구인구직</a>
    </li>
</ul>
```

카테고리컬럼추가

컨트롤러/리포지터리
서비스에 추가코드삽입

DB내의 모든글을가져왔던
방식대신 category값과
일치하는 글만 받아옴

Sql의 where문 과 비슷
Ex) select *from Board
Where category=??

페이징

코드예시

```
package com.bootlec.sbb.board;

import java.util.List;

@GetMapping("/category/notification")
public String notilist(Model model, @RequestParam(value="page",defaultValue="0") int page) {
    Page<Board> paging = this.boardService.getNotiList(page);
    model.addAttribute("paging",paging);
    return "board_list";
}

    public Page<Board> getNotiList(int page) {
        List<Sort.Order> sorts = new ArrayList<>();
        sorts.add(Sort.Order.desc("createDate"));
        Pageable pageable = PageRequest.of(page, 10,Sort.by(sorts));
        return this.boardRepository.findByCategory(pageable, "공지");
    }
```

구현결과

Run | Run Selected | Auto complete | Clear | SQL statement:

```
SELECT * FROM BOARD
```

8	내용무	2024-05-17 10:21:33.929244	null	테스트 데이터입니다.[005]	null	null	null	
9	내용무	2024-05-17 10:21:33.93027	null	테스트 데이터입니다.[006]	null	null	null	
10	내용무	2024-05-17 10:21:33.931253	null	테스트 데이터입니다.[007]	null	null	null	
11	내용무	2024-05-17 10:21:33.932237	null	테스트 데이터입니다.[008]	null	null	null	
12	내용무	2024-05-17 10:21:33.934231	null	테스트 데이터입니다.[009]	null	null	null	
13	내용무	2024-05-17 10:21:33.935228	null	테스트 데이터입니다.[010]	null	null	null	
14	내용무	2024-05-17 10:21:33.937234	null	테스트 데이터입니다.[011]	null	null	null	
15	내용무	2024-05-17 10:21:33.939217	null	테스트 데이터입니다.[012]	null			
16	내용무	2024-05-17 10:21:33.942209	null	테스트 데이터입니다.[013]	null			
17	내용무	2024-05-17 10:21:33.943207	null	테스트 데이터입니다.[014]	null			
18	내용무	2024-05-17 10:21:33.944204	null	테스트 데이터입니다.[015]	null			
19	내용무	2024-05-17 10:21:33.946199	null	테스트 데이터입니다.[016]	null			
20	내용무	2024-05-17 10:21:33.948194	null	테스트 데이터입니다.[017]	null			
21	내용무	2024-05-17 10:21:33.949191	null	테스트 데이터입니다.[018]	null			
22	내용무	2024-05-17 10:21:33.950188	null	테스트 데이터입니다.[019]	null			
23	내용무	2024-05-17 10:21:33.951185	null	테스트 데이터입니다.[020]	null			
24	오늘도 하루가 지나갔다. 또 지나갔다.	2024-05-17 12:39:57.09718	null	나의 하루	1			
25	새로운 내용입니다.	2024-05-20 11:52:22.608622	null	새로운 글입니다.	1			
26	잘 부탁 드립니다.	2024-05-20 11:53:31.463277	null	새로운 회원입니다.	3			
27	qqweqwe	2024-05-21 15:41:23.979215	1234@5678.com	aaaaaq	1			
28	잘 부탁 드립니다.	2024-05-21 15:43:21.525118	2345@6789.com	가입인사 드립니다.	4		null	
31	공지글입니다.	2024-05-22 12:19:40.223338	1234@5678.com	공지합니다!	1	2024-05-22 14:17:58.718741	공지	
32	냉무	2024-05-22 12:28:46.876135	2345@6789.com	일자리 구합니다!	4	null	구인구직	
33	냉무	2024-05-23 11:15:53.913048	1234@5678.com	새로운 공지입니다.(2)	1	null	공지	
34	냉무	2024-05-23 11:16:16.487065	1234@5678.com	새로운 공지입니다.(3)	1	2024-05-23 11:16:24.333472	공지	
35	냉무	2024-05-23 11:17:06.985129	2345@6789.com	알바구합니다.	4	null	구인구직	
36	냉무	2024-05-23 11:17:32.100428	2345@6789.com	새로운 직원을 뽑습니다.	4	null	구인구직	

(34 rows, 9 ms)

Run | Run Selected | Auto complete | Clear | SQL statement:

```
SELECT * FROM BOARD where category = '공지'
```

SELECT * FROM BOARD where category = '공지';

ID	CONTENT	CREATE_DATE	EMAIL	SUBJECT	AUTHOR_ID	MODIFY_DATE	CATEGORY
31	공지글입니다.	2024-05-22 12:19:40.223338	1234@5678.com	공지합니다!	1	2024-05-22 14:17:58.718741	공지
33	냉무	2024-05-23 11:15:53.913048	1234@5678.com	새로운 공지입니다.(2)	1	null	공지
34	냉무	2024-05-23 11:16:16.487065	1234@5678.com	새로운 공지입니다.(3)	1	2024-05-23 11:16:24.333472	공지

(3 rows, 1 ms)

카테고리

구현결과

새 글 등록하기

찾기

전체 공지 구인구직

새 글 등록하기

찾기

전체 공지 구인구직

번호	제목	작성일시	email	작성자
3	[공지] 새로운 공지입니다.(3)	2024-05-23 11:16	1234@5678.com	jsj
2	[공지] 새로운 공지입니다.(2)	2024-05-23 11:15	1234@5678.com	jsj
1	[공지] 공지합니다!	2024-05-22 12:19	1234@5678.com	jsj

이전 1 다음

26	새로운 회원입니다.	2024-05-20 11:53	admin
25	새로운 글입니다. 1	2024-05-20 11:52	jsj

이전 1 2 3 4 다음

기능추가

코드예시

```
@PreAuthorize("isAuthenticated()")
    @GetMapping("/create")
      public String noticeCreate(NoticeForm noticeForm) {
```

```
<div class="col-6" >
   <a th:href="@{/n/create}" class="btn btn-primary"
               sec:authorize="isAuthenticated()"
               th:if="${#authentication.getPrincipal().getUsername() == 'admin'}"
               th:text="공지"></a>
```

```
List<Notice> notices = noticeService.getList(kw);
model.addAttribute("notices", notices);
model.addAttribute("paging", paging);
model.addAttribute("kw", kw);
```

공지사항

admin권한을 확인한다

Admin이아니면 작성버튼이 보이지않는다

조건부여

코드예시

```
  <span th:text="'[공지]'"></span></td>
  <td>
     <span th:style="'display: inline-block; width: 80%; '">
    <a th:href="@{|/n/detail/${notice.id}|}">
        <strong th:text="${notice.subject}"></strong>
    </a>
</span>
```

```
+ "and n.createDate in ("
            + "   select n2.createDate "
            + "   from Notice n2 "
            + "   ORDER BY n2.createDate DESC "
            + "   LIMIT 2)"
            + "order by n.createDate desc")
```

공지사항

admin권한을 확인한다

Admin이아니면 작성버튼이 보이지않는다

항상 게시판위에 출력

글과 별개처리해준다

조건부여

구현결과

```
  <span th:text="'[공지]'"></span></td>
  <td>
     <span th:style="'display: inline-block; width: 80%; '">
    <a th:href="@{|/n/detail/${notice.id}|}">
        <strong th:text="${notice.subject}"></strong>
    </a>
</span>
```

공지사항

admin권한을 확인한다

Admin이아니면 작성버튼이 보이지않는다

항상 게시판위에 출력

글과 별개처리해준다

번호	제목	글쓴이	작성일시
[공지]	4차공지	admin	2024-05-22 15:53
[공지]	3번째공지	admin	2024-05-22 12:16
[공지]	2차공지작성	admin	2024-05-22 12:02

로그인/보안

스프링시큐리티

스프링 하위프레임워크

인증을통한신원확인

프로세스를통한권한관리

두가지기능을통한 CRUD관리

LOGO

로그인

코드예시

```
public SiteUser create(String username, String email, String password) {
        SiteUser user = new SiteUser();
try {
        userService.create(userCreateForm.getUsername(),
        userCreateForm.getEmail(), userCreateForm.getPassword1());
        }catch(DataIntegrityViolationException e) {
            e.printStackTrace();
            bindingResult.reject("signupFailed", "이미 등록된 사용자입니다.");
            return "signup_form";
        }catch(Exception e) {
            e.printStackTrace();
            bindingResult.reject("signupFailed", e.getMessage());
            return "signup_form";
        }
```

회원가입

otPasswordEncoder
로 hash함수사용

가입방지

후 마무리

권한관리

코드예시

권한처리

로그인을 하지않으면

```
@PreAuthorize("isAuthenticated()")
        @GetMapping("/delete/{id}")
        public String questionDelete(Principal principal, @PathVariable("id") Integer id) {
            Question question = this.questionService.getQuestion(id);
            if (!question.getAuthor().getUsername().equals(principal.getName())) {
                throw new ResponseStatusException(HttpStatus.BAD_REQUEST, "삭제권한이 없습니다.");
            }
            this.questionService.delete(question);
            return "redirect:/";
        }
```

느낀점

이지환

기존에 공부했던 방식과 다른코드를 사용하여 비교하면서 공부가되었던것같다

이원진

에러 404와 500을 자주 마주하다보니 맵핑/오류처리등의 중요성을 느낄수 있었습니다.

추형욱

스프링 부트만의 장점 매력을 느낄수있었다 스프링시큐리티의 장점 톰캣/DB등이 내장되어있어서 팀프로젝트에서도 큰문제가생기지않았다

장순재

Boot방식으로 게시판을 만들어 보면서 많은차이를 느낄수있었다

한형빈

h2 db연동을 통해 entity기반의 DB 관리 방식을 배우게 되었다.또한 게시판의 CURD 기능을 구현해보며 스프링 부트의 작업 방식을 경험할 수 있었다.

LOGO


Thank YOU
잘들어주셔서 감사합니다