

Animal Forest

동숲놀이터 커뮤니티 프로젝트

Editor - YUN JAE PIL



목차

1. **프로젝트 개요**
2. **개발환경 및 일정**
3. **화면 및 주요기능 소개**
4. **구현 작업 소스코드**

< 배포 웹사이트 주소 >

<http://52.231.206.1:8080/animalForest/main.do>

* 배포 웹 사이트는 시일 경과 후 종료될 수 있습니다.



프로젝트 개요 - 주제 및 역할

동물의 숲 및 반려동물 정보나눔 커뮤니티

주제

스프링 프레임워크, MVC 를 활용한 동물의숲/반려동물 커뮤니티

주요 기능

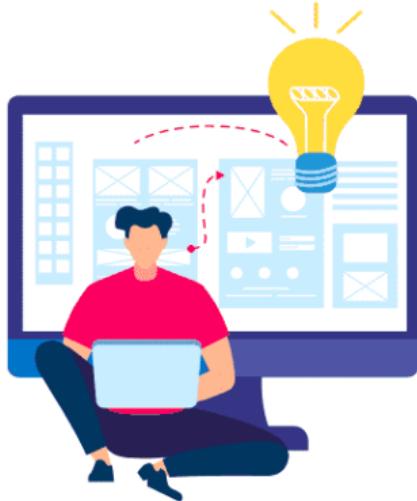
- 최초 회원가입 이후 네이버API, 카카오 API 를 활용한 로그인 기능 제공
- 유튜브 영상 및 이미지를 포함한 게시글을 작성하고 댓글 소통할 수 있는 기능 제공
- 회원간 소모임 주최 / 모집, 참여가 가능하도록 구현
- 취득한 누적포인트에 따른 NEW, FAMILY, VIP, VVIP 등급 아이콘 적용

역할 및 담당

- 메이븐개발환경설정(pom.xml/web.xml/log4j)
- 이미지upload 및 유튜브재생 가능한 자유게시판구현
- 댓글 및 대댓글 및 게시물좋아요(♡)기능구현
- MyPage 프로필 수정 기능 구현

프로젝트 개요 - 개발 도구

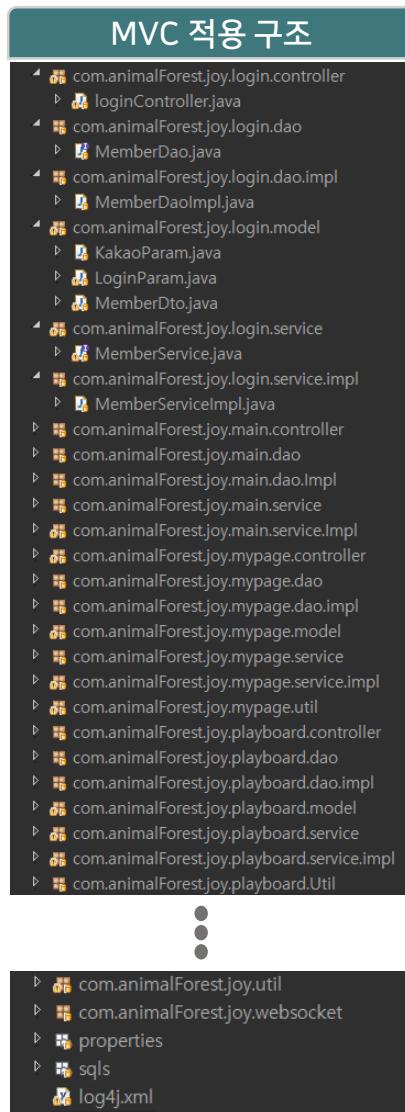
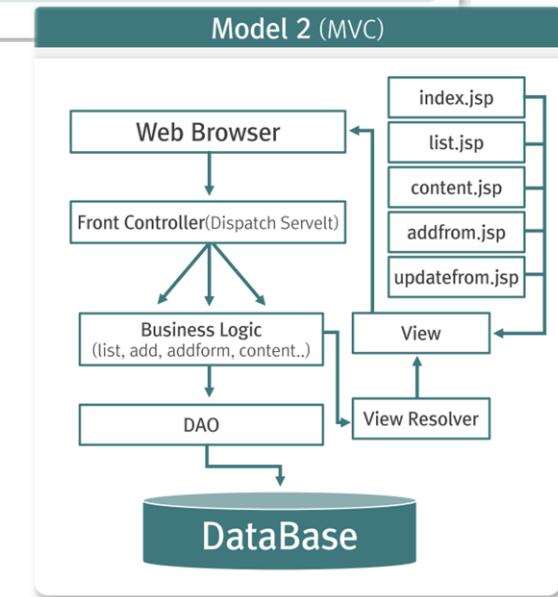
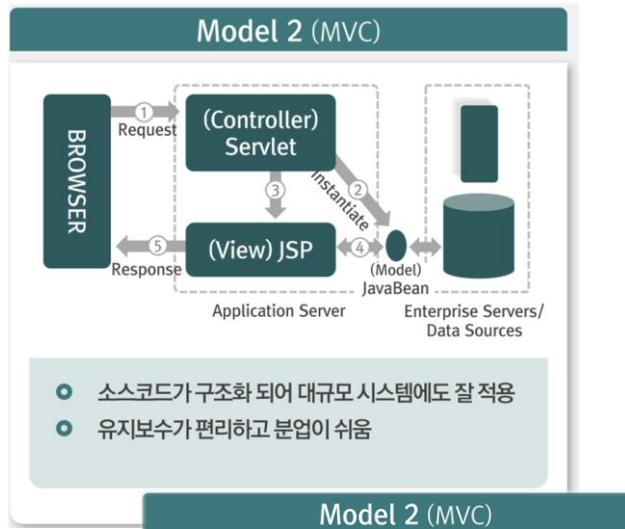
동물의 숲 및 반려동물 정보나눔 커뮤니티



- OS / BROWSER – Windows server 2016 / Google chrome
- Language – Java, JSP, Javascript, Jquery, Oracle SQL, HTML, CSS
- Web Application Server – Tomcat 8.5
- Tool – Spring Tool Suite, VS_code, GitHub, SqlDeveloper
- DB – Oracle 11g
- JDK – 1.8

프로젝트 개요 - 스프링 프레임워크 적용

동물의 숲 및 반려동물 정보나눔 커뮤니티



적용 방식

데이터전달 : DTO 클래스 활용.
 컨트롤러
 서비스 - 서비스impl
 DAO - DAOimpl
 Model (DTO)

패키지를 구분한 형식으로 정리.

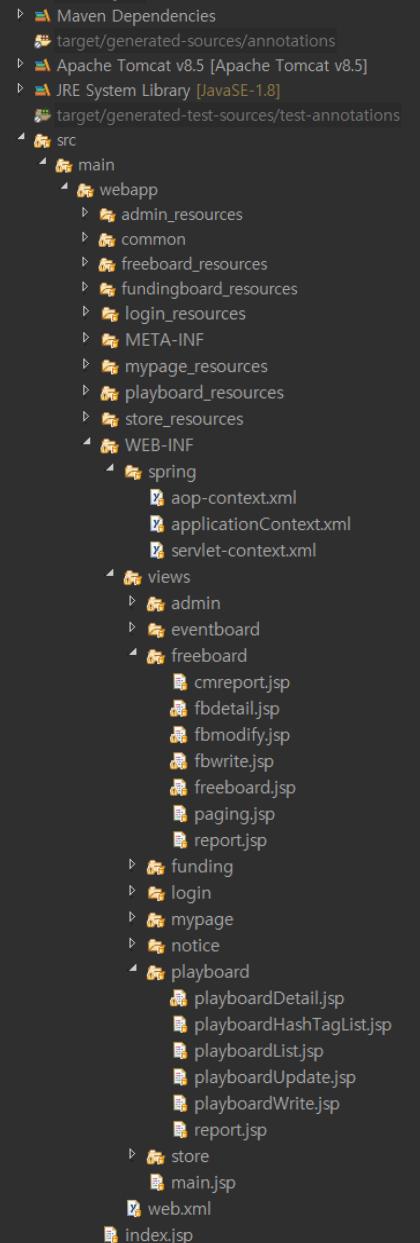


View 구조

View폴더를 목적별 분류.

CSS / JS / images 는
 '목적_리소스' 폴더에 배치.

View 구성



프로젝트 일정

동물의 숲 및 반려동물 정보나눔 커뮤니티

2020.6월 말

프로젝트기획 및
개발목표일정수립



2020. 7월 27일

프로젝트 발표 및 교육수료

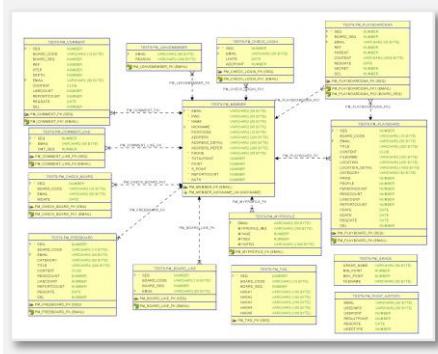


2020. 8월 초

백로그 및
개선요청사항수정적용

2020.07 초

```
animalForest/pom.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2
3<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd" modelVersion="4.0.0">
4
5<groupId>com.company</groupId>
6<artifactId>animalForest</artifactId>
7<name>animalForest Maven Webapp</name>
8<packaging>war</packaging>
9<version>1.0.1-SNAPSHOT</version>
10
11<properties>
12<java-version>1.8</java-version>
13
14<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
15<maven.compiler.source>1.8</maven.compiler.source>
16<maven.compiler.target>1.8</maven.compiler.target>
17<org.springframework-version>4.0.4.RELEASE</org.springframework-version>
18<org.aspectj-version>1.7.3</org.aspectj-version>
19<org.slf4j-version>1.7.25</org.slf4j-version>
20
21</properties>
```



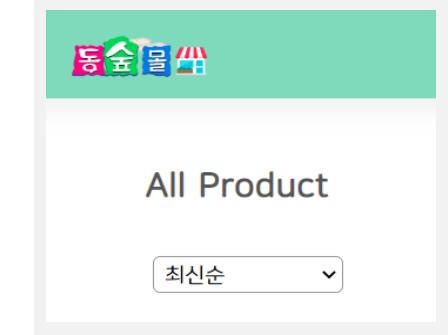
환경세팅 및 DB구성 / 개발작업착수

2020.7월 말



커뮤니티완성버전웹배포

이후일정



All Product

최신순

쇼핑몰 모듈구현 고려

프로젝트 화면 및 기능 소개(1/6)

동물의 숲 및 반려동물 정보나눔 커뮤니티

회원가입

test999@naver.com 1 이메일인증

213399 인증확인

.....

.....

불일치

동숲마스터 사용가능

홍길동

07803 1 우편번호검색

서울 강서구 강서로 375

999-999

(마곡동)

* 연락처(13자리이내)

회원가입

회원가입 및 로그인페이지

- 모바일 웹 접속을 고려한 UI

< 회원가입

- 1 이메일 인증번호 발송 기능 및 DAUM 우편번호 검색 API 적용.

로그인 >

- 2 이메일 기반 회원정보 확인, 카카오, 네이버 로그인 지원.

회원가입

로그인

Email

Password

ID저장

비밀번호 찾기

로그인

카카오 로그인

N 네이버 아이디로 로그인

카카오계정 - Chrome accounts.kakao.com/login?continue=https%3A%2F%2Fka...

kakao

카카오계정 (이메일 또는 전화번호)

비밀번호

로그인 상태 유지

로그인

또는

QR코드 로그인

회원가입

카카오계정 | 비밀번호 찾기

프로젝트 화면 및 기능 소개(2/6)

동물의 숲 및 반려동물 정보나눔 커뮤니티

마이페이지 > 프로필 정보(사진 및 소개) 적용

The screenshot shows the Naver Profile Edit page. On the left, there's a form for editing profile details like nickname, age, gender, and introduction. A red box highlights the nickname field ('Jef441') and the introduction field ('등록된 소개가 없습니다22'). A yellow box labeled '1' points to the top right corner of the page, where a dropdown menu is open, showing '마이페이지' (My Page) as the selected item. Another yellow box labeled '2' points to the preview area on the right, which shows a cartoon character's face as the profile picture and the introduction text '등록된 소개가 없습니다22'.

- 1
 - 로그인후 헤더의 닉네임 클릭하여 '마이페이지' 진입 가능
 - 톱니바퀴아이콘을 클릭해 사이드메뉴 뷰/히든화 처리

- 2
 - 프로필 수정완료 시 새로운 프로필과 소개글 저장.
 - 적용된 프로필 이미지는 나의 활동 게시물에 노출됨



프로젝트 화면 및 기능 소개(3/6)

동물의 숲 및 반려동물 정보나눔 커뮤니티

공지사항 및 자유게시판 목록 구현화면

Home **공지** 자유게시판 ▾ 소모임 후원 스토어

1

2

3

동물의 숲 셀을 풀며 다른사람과 소통해보세요

동숲이야기 게시판

동물의 숲 셀을 풀며 다른사람과 소통해보세요

글쓰기

정렬 분류

List

번호	분류	제목	작성자	작성일	조회수	좋아요
3	잡담	동물의 숲에서 가장 귀여운동물은 누구? (0)	제프리0722	2020-07-28	9	0
2	잡담	동물의 숲 애플리 나비로벳따우 영상 (7)	Jef441	2020-07-27	26	1
1	자랑	첫번째 인사 써봅니다. (2)	Jef441	2020-07-20	25	1

선택 검색

1

네비게이션바 '공지' 클릭 시 공지사항 게시판 이동

- 글쓰기 기능은 운영자 권한자만 가능.

AnimalForest

공지사항

동술놀이터에서 알립니다!

전체 1

No.	구분	제목
1	자유게시판 공지	자유 공지사항 첫번째입니다.

구분 제목

일 조회수

« 1 »

2

자유게시판 3개 카테고리(주제) 구성

3

List 상세화면-정렬 및 검색/페이지 기능 적용

- [최신순/좋아요/조회수] 기준으로 정렬 가능
- [제목/내용/닉네임] 검색 가능
- 페이지 적용

프로젝트 화면 및 기능 소개(4/6)

동물의 숲 및 반려동물 정보나눔 커뮤니티

1

유튜브 비디오 삽입

여기 embed 코드를 붙여넣으세요

또는

유튜브 주소(URL)을 붙여넣으세요

너비: 640 높이: 360

한 줄로 네비 (일렬한 네비와 높이를 뿐시하고 정 너비에 맞춤) 비디오 이미지와 콩크만 끌기

비디오 마지막에 축전 영상 보기 엣글 embed 코드 사용

개인정보 보호 모드 활성화 자동 재생

시작 시점 (ss 또는 mm:ss 또는 hh:mm:ss) 플레이어 컨트롤 표시

확인 취소

자유게시판 글쓰기 / 수정하기 상세화면

1 CKEditor 4.0 기본 플러그인 + YouTube 플러그인 적용

▼ 적용 예시) 영상 포함 게시글

동물의 숲 애플이 나비보벳따우 영상

작성일: 2020-07-27 09:20 좋아요: 2 조회수: 27

[모든 숲] 동물의 숲 대표 아이돌 애플이의 나비보벳따우

나중에 시... 자유

또 마이크를 보는 애플이!! (기대)

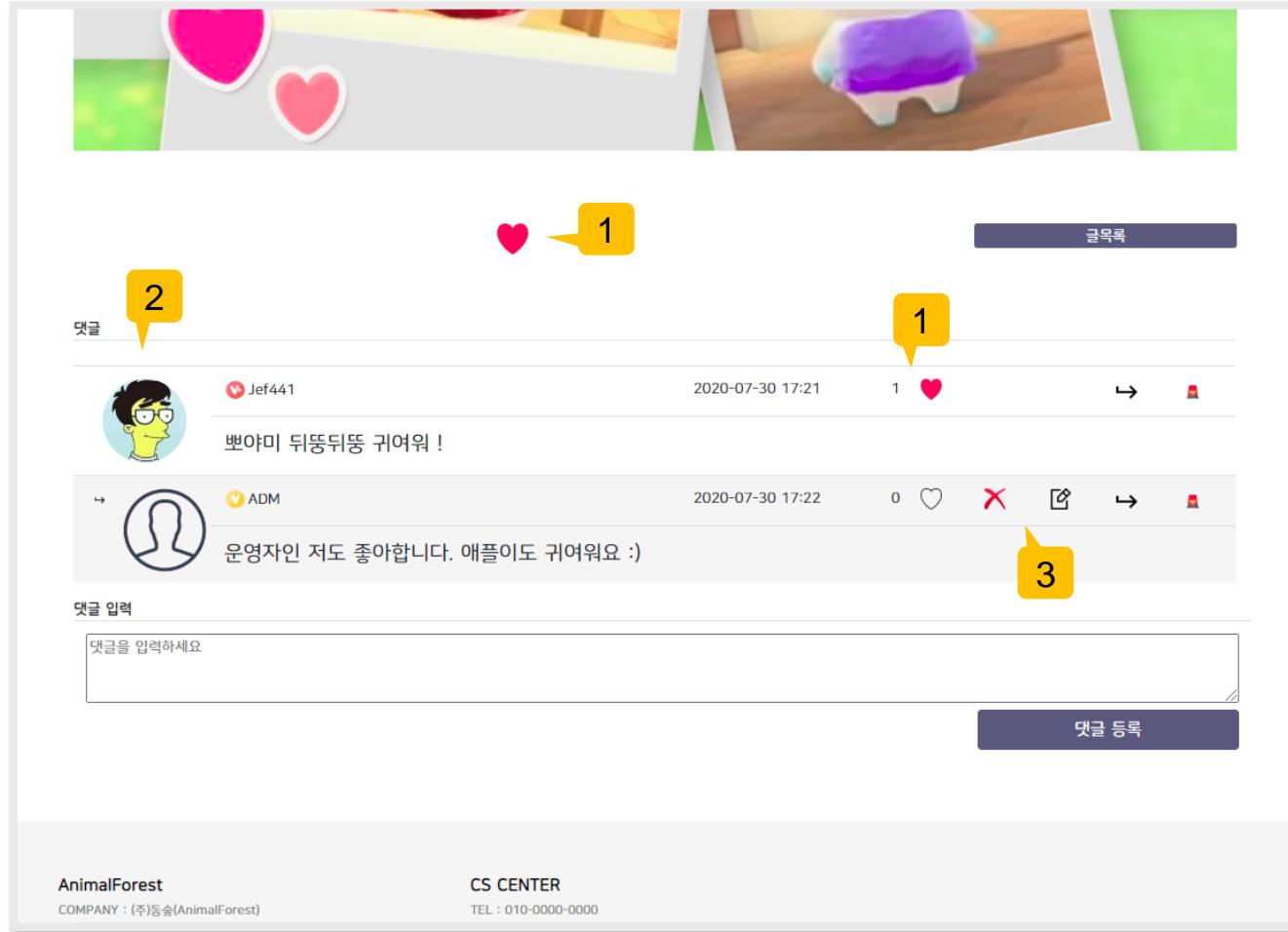
나비보벳따우~

3:02 / 5:11

YouTube

프로젝트 화면 및 기능 소개(5/6)

동물의 숲 및 반려동물 정보나눔 커뮤니티



게시글 좋아요 및 댓글 / 대댓글 기능

- 1 게시물과 댓글에 좋아요 버튼 구현/ 누적 좋아요 개수 표시

작성일: 2020-07-28 17:20 좋아요:2 조회수:13

- 게시글 본문의 누적 좋아요 갯수는 게시글 상단에 표현

- 2 게시물 하단 댓글 등록/대댓글 작성 기능

- 3 댓글의 작성자에겐 수정 및 삭제 기능 노출



프로젝트 화면 및 기능 소개(6/6)

동물의 숲 및 반려동물 정보나눔 커뮤니티

소모임

동숲놀이터에서 반려동물에 대한 정보를 공유할 수 있는 친구, 나의 반려동물의 친구를 만나보세요!

1 소모임 만들기

모임유형

검색분류

검색

최신순

1 [산책] 대구 경북 애견동반 산책길
2020년 8월 22일 (토)
경북 경산시
0
0명 참여중 모집인원 5명

2 [산책] 도봉산역 산책하실분
2020년 7월 31일 (금)
서울 도봉구
1
1명 참여중 모집인원 5명

소모임 생성 및 참여 기능

- 1 유저가 직접 소모임 모집기한과 장소를 정해 올릴 수 있음
- 2 모집 마감일/모임 예정일/모집인원 등의 조건에 따라 참여신청 가능 여부가 달라짐.

모임 주최자

ADM

모임 예정일

2020년 7월 31일 (금) 2

모집 마감일

2020년 7월 30일 (목)

모임 장소

서울 도봉구 도봉로 948

[산책] 도봉산역 산책하실분
2020년 7월 31일 (금)
서울 도봉구
1
2명 참여중 모집인원 5명 2

#도봉산 #산책

참여 취소

주요 구현 작업 요약

동물의 숲 및 반려동물 정보나눔 커뮤니티

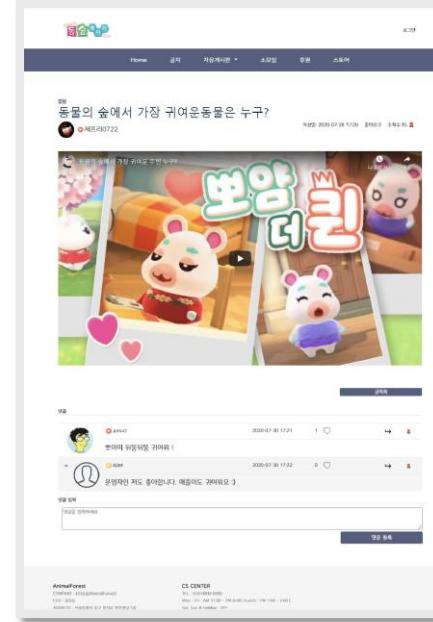
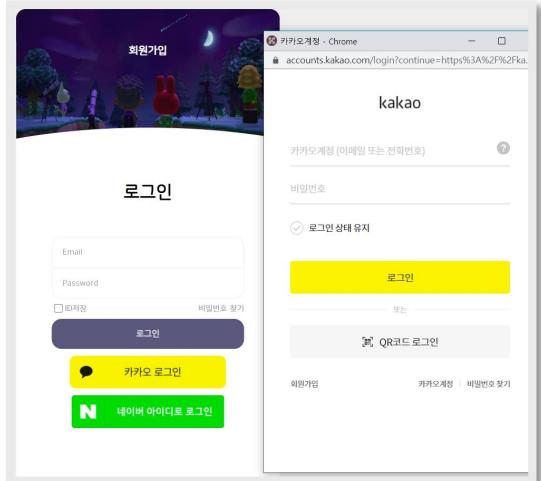
■ SPRING 메이븐 공통 설정(pom.xml / web.xml / log4j)

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

  <groupId>com.company</groupId>
  <artifactId>animalForest</artifactId>
  <name>animalForest Maven Webapp</name>
  <packaging>war</packaging>
  <version>1.0.1-SNAPSHOT</version>

  <properties>
    <java-version>1.8</java-version>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>1.7</maven.compiler.source>
    <maven.compiler.target>1.7</maven.compiler.target>
    <org.springframework-version>4.0.4.RELEASE</org.springframework-version>
    <org.aspectj-version>1.7.3</org.aspectj-version>
    <org.slf4j-version>1.7.25</org.slf4j-version>
  </properties>
```

■ 로그인 로직 및 뷰페이지



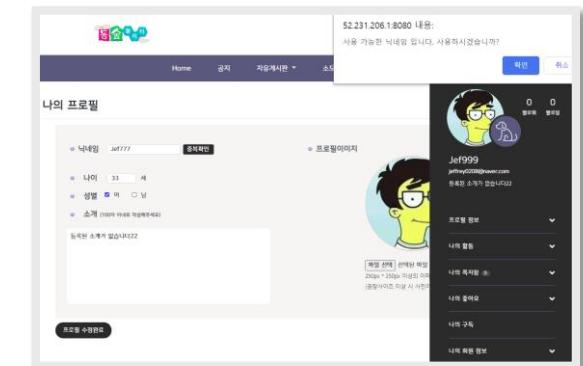
■ MyPage 프로필 설정

■ 자유게시판 글쓰기

- 글쓰기/수정하기/삭제하기
- 좋아요 누르기

■ 댓글/대댓글

- 댓글쓰기/수정하기/삭제하기
- 좋아요 누르기



작업 소스코드 - 공통설정(1/2)

공통 설정 로그작업 – Interceptor / Aspect 적용하기

```

=====
START =====/
Request URI : /animalForest/freeboard_boarddetail.do
==> 1 Controller : com.animalForest.joy.freeboard.controller.FreeboardController.freeboard_boarddetail()
==> 2 ServiceImpl : com.animalForest.joy.freeboard.service.impl.FreeboardServiceImpl.dolikecheck()
==> 3 DaoImpl : com.animalForest.joy.freeboard.dao.impl.FreeboardDaoImpl.dolikecheck()
    QueryId : freeboard.dolikecheck
    들어온 값 : boardlikeDto [seq=0, board_code=AFTSTORY, board_seq=122, email=]
2020-07-31 00:01:44,923 INFO === SQL : SELECT COUNT(*) FROM PM_BOARD_LIKE
WHERE EMAIL = "AND BOARD_SEQ = 122"
2020-07-31 00:01:44,926 INFO [jdbc.resultsettable] |-----|
2020-07-31 00:01:44,927 INFO [jdbc.resultsettable] |COUNT(*)|
2020-07-31 00:01:44,927 INFO [jdbc.resultsettable] |-----|
2020-07-31 00:01:44,927 INFO [jdbc.resultsettable] |0      |
2020-07-31 00:01:44,927 INFO [jdbc.resultsettable] |-----|
==> 2 ServiceImpl : com.animalForest.joy.freeboard.service.impl.FreeboardServiceImpl.getdetail()
==> 3 DaoImpl : com.animalForest.joy.freeboard.dao.impl.FreeboardDaoImpl.getdetail()
    QueryId : freeboard.getdetail
    들어온 값 : 122
2020-07-31 00:01:44,927 INFO === SQL : SELECT F.BOARD_CODE,
F.SEQ,
F.EMAIL,
F.CATEGORY,
F.TITLE,
F.CONTENT,
F.READCOUNT,
=====
INNER JOIN PM_MEMBER b
ON a.EMAIL = b.EMAIL
LEFT OUTER JOIN PM_MYPROFILE c
ON a.EMAIL = c.EMAIL )F
WHERE F.BOARD_SEQ = 122
ORDER BY REF ASC, STEP ASC
2020-07-31 00:01:44,933 INFO [jdbc.resultsettable] |-----|-----|-----|-----|-----|-----|
2020-07-31 00:01:44,933 INFO [jdbc.resultsettable] |SEQ |BOARD_CODE |BOARD_SEQ |REF |STEP |DEPTH
2020-07-31 00:01:44,933 INFO [jdbc.resultsettable] |---|-----|-----|-----|-----|-----|
2020-07-31 00:01:44,933 INFO [jdbc.resultsettable] |128 |AFTSTORY |122   |18  |0   |jeffrey020
2020-07-31 00:01:44,933 INFO [jdbc.resultsettable] |129 |AFTSTORY |122   |18  |1   |admin1@r
2020-07-31 00:01:44,933 INFO [jdbc.resultsettable] |-----|-----|-----|-----|-----|-----|
==> 2 ServiceImpl : com.animalForest.joy.freeboard.service.impl.FreeboardServiceImpl.readcountup()
==> 3 DaoImpl : com.animalForest.joy.freeboard.dao.impl.FreeboardDaoImpl.readcountup()
    QueryId : freeboard.readcountup
    들어온 값 : 122
2020-07-31 00:01:44,934 INFO === SQL : UPDATE PM_FREEBOARD
SET READCOUNT = READCOUNT + 1
WHERE SEQ = 122
=====
END =====/

```

인터셉터 preHandle / postHandle 활용

```

1 MemberDaoImpl.java 2 LoggerInterceptor.java x
1 package com.animalForest.joy.util;
2
3 import javax.servlet.http.HttpServletRequest;
4
5 public class LoggerInterceptor extends HandlerInterceptorAdapter {
6     protected Log log = LogFactory.getLog(LoggerInterceptor.class);
7
8     @Override
9     public boolean preHandle(HttpServletRequest request, HttpServletResponse response,
10         Object handler) throws Exception {
11
12         System.out.println();
13         System.out.println("===== START =====/");
14         System.out.println(" Request URI \t: " + request.getRequestURI());
15         return super.preHandle(request, response, handler);
16     }
17
18     @Override
19     public void postHandle(HttpServletRequest request, HttpServletResponse response,
20         Object handler, ModelAndView modelAndView) throws Exception {
21
22         System.out.println("===== END =====/");
23         System.out.println();
24     }
25
26
27
28
29
30
31
32
33
34

```

AOP @Around 활용

```

1 package com.animalForest.joy.util;
2
3 import org.apache.commons.logging.Log;
4
5 @Aspect
6 public class LoggerAspect {
7     protected Log log = LogFactory.getLog(LoggerAspect.class);
8     static String name="";
9     static String type="";
10    static String method="";
11
12    @Around("execution(* com.animalForest.joy..*(..))")
13    public Object logPrint(ProceedingJoinPoint joinPoint) throws Throwable{
14        type = joinPoint.getSignature().getDeclaringTypeName();
15        method = joinPoint.getSignature().getName()+"()";
16
17        if(type.indexOf("controller")>-1) {
18            name="==> 1 Controller \t: ";
19            System.out.println(name+type+"."+method);
20        }
21        else if(type.indexOf("service")>-1) {
22            name="==> 2 ServiceImpl \t: ";
23            System.out.println(name+type+"."+method);
24        }
25        else if(type.indexOf("dao")>-1) {
26            name="==> 3 DaoImpl : \t: ";
27            System.out.println(name+type+"."+method);
28        }
29        else {
30            name="==> 0 기타: \t: ";
31        }
32
33        return joinPoint.proceed();
34    }
35
36
37
38

```



작업 소스코드 - 공통설정(2/2)

공통 설정 로그작업 – DAO 작업과정 및 마이바티스 쿼리 흐름 적용하기

```

=====
START =====
Request URI : /animalForest/freeboard_boarddetail.do
==> 1 Controller : com.animalForest.joy.freeboard.controller.FreeboardController.freeboard_boarddetail()
==> 2 ServiceImpl : com.animalForest.joy.freeboard.service.impl.FreeboardServiceImpl.dolikecheck()
==> 3 DaoImpl : com.animalForest.joy.freeboard.dao.impl.FreeboardDaoImpl.dolikecheck()
    QueryId : freeboard.dolikecheck
    들어온 값 : boardlikeDto [seq=0, board_code=AFTSTORY, board_seq=122, email=]
2020-07-31 00:01:44,923 INFO === SQL : SELECT COUNT(*) FROM PM_BOARD_LIKE
WHERE EMAIL = "AND BOARD_SEQ = 122"
2020-07-31 00:01:44,926 INFO [jdbc.resultsettable] |-----|
2020-07-31 00:01:44,927 INFO [jdbc.resultsettable] |COUNT(*)|
2020-07-31 00:01:44,927 INFO [jdbc.resultsettable] |-----|
2020-07-31 00:01:44,927 INFO [jdbc.resultsettable] |0   |
2020-07-31 00:01:44,927 INFO [jdbc.resultsettable] |-----|
==> 2 ServiceImpl : com.animalForest.joy.freeboard.service.impl.FreeboardServiceImpl.getdetail()
==> 3 DaoImpl : com.animalForest.joy.freeboard.dao.impl.FreeboardDaoImpl.getdetail()
    QueryId : freeboard.getdetail
    들어온 값 : 122
2020-07-31 00:01:44,927 INFO === SQL : SELECT F.BOARD_CODE,
F.SEQ,
F.EMAIL,
F.CATEGORY,
F.TITLE,
F.CONTENT,
F.READCOUNT,
=====
INNER JOIN PM_MEMBER b
ON a.EMAIL = b.EMAIL
LEFT OUTER JOIN PM_MYPROFILE c
ON a.EMAIL = c.EMAIL )F
WHERE F.BOARD_SEQ = 122
ORDER BY REF ASC, STEP ASC
2020-07-31 00:01:44,933 INFO [jdbc.resultsettable] |-----|-----|-----|-----|-----|-----|
2020-07-31 00:01:44,933 INFO [jdbc.resultsettable] |SEQ |BOARD_CODE |BOARD_SEQ |REF |STEP |DEPTH
2020-07-31 00:01:44,933 INFO [jdbc.resultsettable] |-----|-----|-----|-----|-----|-----|
2020-07-31 00:01:44,933 INFO [jdbc.resultsettable] |128 |AFTSTORY |122 |18 |0 |0 |jeffrey02@r
2020-07-31 00:01:44,933 INFO [jdbc.resultsettable] |129 |AFTSTORY |122 |18 |1 |1 |admin1@r
2020-07-31 00:01:44,933 INFO [jdbc.resultsettable] |-----|-----|-----|-----|-----|-----|
==> 2 ServiceImpl : com.animalForest.joy.freeboard.service.impl.FreeboardServiceImpl.readcountup()
==> 3 DaoImpl : com.animalForest.joy.freeboard.dao.impl.FreeboardDaoImpl.readcountup()
    QueryId : freeboard.readcountup
    들어온 값 : 122
2020-07-31 00:01:44,934 INFO === SQL : UPDATE PM_FREEBOARD
SET READCOUNT = READCOUNT + 1
WHERE SEQ = 122
=====
END =====

```

4

5

4 Dependency&dataSource&log4j.xml 설정

```

<!-- Logging -->
<dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>slf4j-api</artifactId>
    <version>${org.slf4j-version}</version>
</dependency>
<dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>jcl-over-slf4j</artifactId>
    <version>${org.slf4j-version}</version>
</dependency>
<dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>slf4j-log4j12</artifactId>
    <version>${org.slf4j-version}</version>
</dependency>
<dependency>
    <groupId>log4j</groupId>
    <artifactId>log4j</artifactId>
    <version>1.3alpha-8</version>
</dependency>
<dependency>
    <groupId>org.lazyluke</groupId>
    <artifactId>log4jdbc-remix</artifactId>
    <version>0.2.7</version>
</dependency>

```

```

<bean id="dataSource" class="net.sf.Log4Jdbc.Log4JdbcProxyDataSource">
    <constructor-arg ref="dataSourceSpied" />
    <property name="logFormatter">
        <bean class="net.sf.log4jdbc.tools.Log4JdbcCustomFormatter">
            <property name="LoggingType" value="MULTI_LINE" />
            <property name="sqlPrefix" value="==== SQL : " />
        </bean>
    </property>
</bean>

```

```

LoggerInterceptor.java  LoggerAspect.java  log4.xml *
23
24     <!-- Query Loggers -->
25● <logger name="jdbc.sqlOnly" additivity="false">
26     <level value="INFO"/>
27     <appender-ref ref="console-infolog"/>
28 </logger>
29
30● <logger name="jdbc.resultsettable" additivity="false">
31     <level value="INFO"/>
32     <appender-ref ref="console"/>
33 </logger>
34

```

DAO구현 시 쿼리id와 입력값이 있는 경우 출력되도록 적용

```

MemberDaoImpl.java x
1 package com.animalForest.joy.login.dao.impl;
2
3 import org.apache.ibatis.session.SqlSession;
4
5 @Repository
6 public class MemberDaoImpl implements MemberDao {
7
8     @Autowired
9     SqlSession sqlSession;
10
11     String ns = "Member.";
12
13     protected void printQueryId(String queryId) {
14         System.out.println("\tQueryId \t: " + queryId);
15     }
16
17     protected void printQueryId(String queryId, Object object) {
18         System.out.println("\tQueryId \t: " + queryId);
19         System.out.println("\t 들어온 값 \t: " + object);
20     }
21
22     @Override
23     public void addMember(MemberDto mem) {
24         printQueryId(ns + "addMember", mem );
25         sqlSession.insert(ns + "addMember", mem);
26     }

```

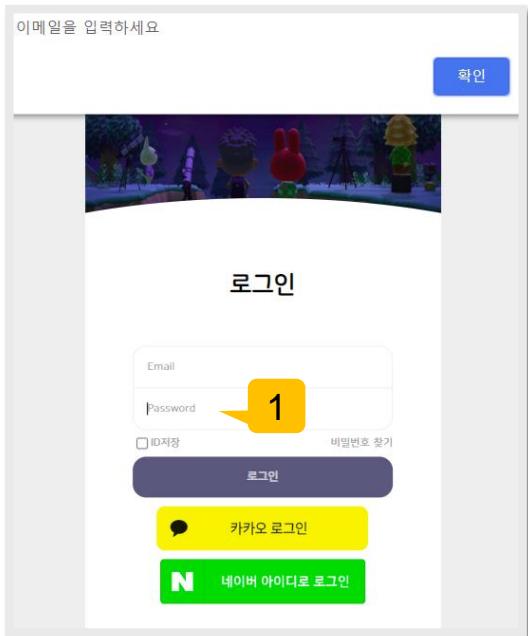


작업 소스코드 - 로그인 뷰 및 로직구현(1/3)

로그인 뷰 및 [로그인] Button 클릭 로직 구현

1 패스워드 입력란에서 Enter 키 입력 시 [로그인] 버튼클릭 효과 적용하기

```
144 $('input[name=pwd]', document.loginform).keydown(function(event) {
145   //로그인 품 패스워드 입력창에 커서가 끌어간 상태에서 엔터키를 누르면 로그인하기 버튼 클릭효과가 나오게 적용
146   if (event.keyCode == '13') {
147     $("#login-btn").trigger("click");
148   }
149 })
150 }) ;
```



2 로그인버튼클릭시 아이디/패스워드검증 흐름

간단히 프론트파트 검증 후
Ajax 활용 요청 전달

3

Ajax을 활용, 로그인 품데이터를
회원검증 컨트롤러로 전달

검증 후, 컨트롤러의 리턴으로

Num=0~8값중 한 값을 받아
success의 해당되는 조건 수행

3

반환과정은 다음 슬라이드 참고

```
loginController.java  login.jsp  input_checkjs x
614 // 로그인 submit
615 $("#login-btn").on("click", function() {
616   var id = $("input[name=email]", document.loginform);
617   var pwd = $("input[name=pwd]", document.loginform);
618
619   if ( id.val() == "" ) {
620     alert("이메일을 입력하세요");
621     id.focus();
622   }
623   else if (pwd.val() == "") {
624     alert("비밀번호를 입력하세요");
625     pwd.focus();
626   }
627   else {
628     var login_data = $("#login-form").serialize();
629
630     $.ajax({
631       url : "loginAf.do",
632       method : "post",
633       data : login_data,
634       dataType : 'json',
635       success : function(num) {
636         console.log("userChk = " + num);
637
638         if ( num == 0 ) { //DB에 아이디가 없는 경우
639           console.log("일치하는 회원 찾지 못함");
640           alert("[id] 비밀번호를 확인해주세요.");
641         }
642         if ( num == 1 ) { //DB에 아이디가 있는 경우 => 로그인
643           console.log("로그인중...");
644
645           //포인트지급관련
646           $.ajax({
647             url : "loginPointCheck.do",
648             method : "post",
649             data : login_data,
650             dataType : 'json',
651             success : function( pointcheck ) {
652
653               if( num == 8 ) {
654                 console.log("관리자계정");
655                 location.href="adminMain.do";
656               }
657             },fail : function(error) {
658               alert(JSON.stringify(error));
659             }
660           });
661
662         }
663       });
664
665     });
666   });
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687 );
```

작업 소스코드 – 로그인 뷰 및 로직구현(2/3)

로그인 DB 데이터 참조 로직

LoginController.java 3-1 : Service>DAO>Mybatis과정(user&leaveCheck 케이스)

```

183● @ResponseBody
184 @RequestMapping(value = "loginAf.do", method = { RequestMethod.GET, RequestMethod.POST })
185 public int loginAf(MemberDto mem, HttpServletRequest req, Model model) {
186     int num;
187     MemberDto user = memService.loginCheck(mem);
188     boolean leaveCheck = memService.leaveMemCheck(mem.getEmail());
189 }
```

3-1

MemberServiceImpl.java



```

45● @Override
46     public MemberDto loginCheck(MemberDto mem) {
47         return memDao.loginCheck(mem);
48     }
49
50● @Override
51     public boolean leaveMemCheck(String email) {
52         return memDao.leaveMemCheck(email);
53     }
```

MemberDaolmpl.java



```

50● @Override
51     public MemberDto loginCheck(MemberDto mem) {
52         printQueryId(ns + "loginCheck", mem );
53         return sqlSession.selectOne(ns + "loginCheck", mem);
54     }
55
56● @Override
57     public boolean leaveMemCheck(String email) {
58         printQueryId( ns + "leaveMemCheck", email);
59         int n = sqlSession.selectOne(ns + "leaveMemCheck", email);
60         return n>0?true:false;
61     }
```

Cf. MemberDto.java

```

1 package com.animalForest.joy.login.model;
2
3 import java.io.Serializable;
4
5 public class MemberDto implements Serializable {
6
7     private String email; // 이메일
8     private String pwd; // 비밀번호
9     private String nickname; // 닉네임
10    private String name; // 이름
11    private String postcode; // 우편번호
12    private String address; // 주소
13    private String address_detail; // 상세주소
14    private String address_refer; // 상세주소 참고항목
15    private String phone; // 연락처
16    private int totalpoint; // 누적포인트
17    private int point; // 현재포인트
18    private int reportcount; // 신고수
19    private int auth; // 회원구분 : 1 일반회원 3 본인인증회원 5 블랙리스트회원 / 8 관리자
20
21    private int savepoint;
22    private int use_point;
```

Member.xml(오라클 쿼리문 적용)

```

26 <!-- 로그인시 회원가입 여부 체크 -->
27 <select id="LoginCheck" parameterType="com.animalForest.joy.Login.model.MemberDto" resultType="com.animalForest.joy.Login.model.MemberDto">
28     SELECT EMAIL, PWD, NICKNAME, NAME, POSTCODE, ADDRESS, ADDRESS_DETAIL, ADDRESS_REFER, PHONE, TOTALPOINT, POINT, S_POINT, REPORTCOUNT, AUTH
29     FROM PM_MEMBER
30     WHERE EMAIL=#{email} AND PWD=#{pwd}
31 </select>
32
33 <!-- 할퇴회원인지 확인하기 -->
34 <select id="LeaveMemCheck" parameterType="java.lang.String" resultType="java.lang.Integer">
35     SELECT COUNT(*) FROM PM_LEAVEMEMBER WHERE EMAIL=#{email}
36 </select>
37
38
```

작업 소스코드 – 로그인 뷰 및 로직구현(3/3)

로그인 컨트롤러, 구분작업 후 반환 과정

LoginController.java 3-2 : 앞선 과정에서 가져온 Data 활용, 회원의 구분값 지정

```

183● @ResponseBody
184 @RequestMapping(value = "loginAf.do", method = { RequestMethod.GET, RequestMethod.POST })
185 public int loginAf(MemberDto mem, HttpServletRequest req, Model model) {
186     int num;
187     MemberDto user = memService.loginCheck(mem);
188     boolean leaveCheck = memService.leaveMemCheck(mem.getEmail());
189
190     if (user == null) {
191         num = 0; // "가입되지 않은 계정";
192     } else {
193         num = 1; // "가입되어 있는 계정";
194     }
195     if (leaveCheck) {
196         num = 2; // "탈퇴한 계정";
197     }
198     if (user != null && user.getReportcount() >= 10) {
199         num = 3; // "신고로 자동 활동정지된 계정";
200     }
201     if (user != null && user.getAuth() == 4) {
202         num = 4; // "관리자에 의한 활동이 정지된 계정";
203     }
204     if (user != null && user.getAuth() == 8) { ← B
205         num = 8; // "관리자 계정";
206         // 로그인시 유저정보 가져오기
207         MemberDto loginuser = memService.getUser(user.getEmail());
208         // 로그인 유저의 프로필 가져오기
209         MyProfileDto profiledto = memService.getLoginUserProfile(user.getEmail());
210         // 로그인 유저의 팻프로필 가져오기
211         MyPetProfileDto mypet = memService.getLoginUserPetProfile(user.getEmail());
212         // 세션생성
213         req.getSession().setAttribute("login", loginuser);
214         System.out.println("loginAf.do>>> 관리자ver 로그인 세션생성");
215
216         req.getSession().setAttribute("userProfile", profiledto);
217         System.out.println("mypagehome.do>>> userProfile 세션생성");
218
219         req.getSession().setAttribute("mypet", mypet);
220         System.out.println("mypagehome.do>>> mypet 세션생성");
221     }
222     return num;
223 }
```

A

: 정상로그인 '회원' 유저인 경우, 아래의 첫점속 포인트 지급을 거친 후 세션 적용

```

@RequestBody
@RequestMapping(value = "loginPointCheck.do", method = { RequestMethod.GET, RequestMethod.POST })
public boolean loginPointCheck(MemberDto user, HttpServletRequest req, Model model) {
    // 로그인 당일 날짜 얻어오기(첫로그인 확인용)
    SimpleDateFormat dateformat = new SimpleDateFormat("yyyy-MM-dd");
    Date time = new Date();
    String todate = dateformat.format(time);
    System.out.println("loginPointCheck todate : " + todate);

    // 당일 첫로그인 포인트지급 여부 확인
    LoginParam param = new LoginParam(user.getEmail(), todate);
    boolean pointcheck = memService.loginPointCheck(param);

    // 당일 첫로그인 이력 존재 -> 로그인 포인트 지급 X
    if (pointcheck) {
        System.out.println("오늘의 첫 로그인이 아닙니다");
    }
    // 당일 첫로그인 이력 없음 -> 로그인 포인트 지급실행
    else if (!pointcheck) {
        System.out.println("오늘 첫로그인입니다");

        // 매일 첫로그인 여부 DB저장, 포인트지급 (일 2번째 로그인부터는 포인트 지급X, 300p 지급)
        param = new LoginParam(user.getEmail(), "", 300);
        boolean b = memService.addLoginPoint(param);

        if (b)
            System.out.println("출석 포인트 지급완료");
        else
            System.out.println("출석 포인트 지급 실패..");
    }
}
```

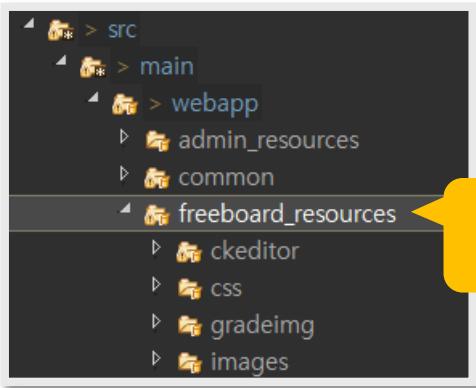
B

: 정상로그인 '어드민' 관리자인 경우, num=8과 함께 세션을 적용

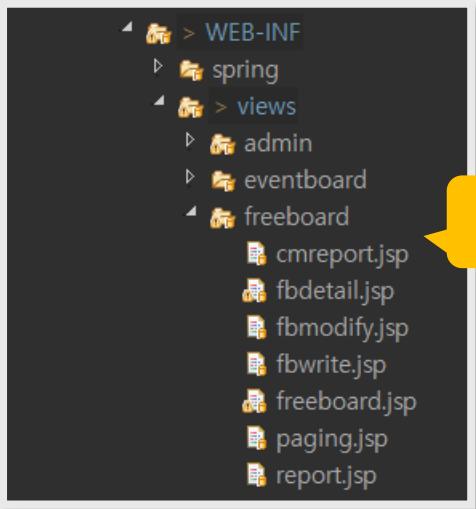
작업 소스코드 – 자유게시판 글쓰기 구현(1/3)

자유게시판 / 댓글 구현 개요

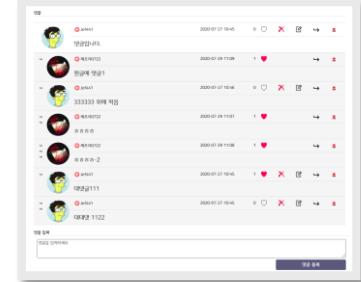
프로젝트 파일 구성



**자유게시판리소스
CSS/이미지 / CKeditor**



자유게시판뷰페이지



```

com.animalForest.joy.freeboard.controller
  FreeboardController.java
com.animalForest.joy.freeboard.dao
  FreeboardDao.java
com.animalForest.joy.freeboard.dao.impl
  FreeboardDaoImpl.java
com.animalForest.joy.freeboard.model
  BoardlikeDto.java
  CmreportDto.java
  CommentDto.java
  CommentlikeDto.java
  FbParam.java
  FreeboardDto.java
  ReportDto.java
com.animalForest.joy.freeboard.service
  FreeboardService.java
com.animalForest.joy.freeboard.service.impl
  FreeboardServiceImpl.java
  
```

자유게시판컨트롤러

**자유게시판DAO 인터페이스
&DAO Implement**

자유게시판활용 모델객체

**자유게시판서비스인터페이스
&서비스Implement**

작업 소스코드 – 자유게시판 글쓰기 구현(2/3)

자유게시판 글쓰기 소스코드

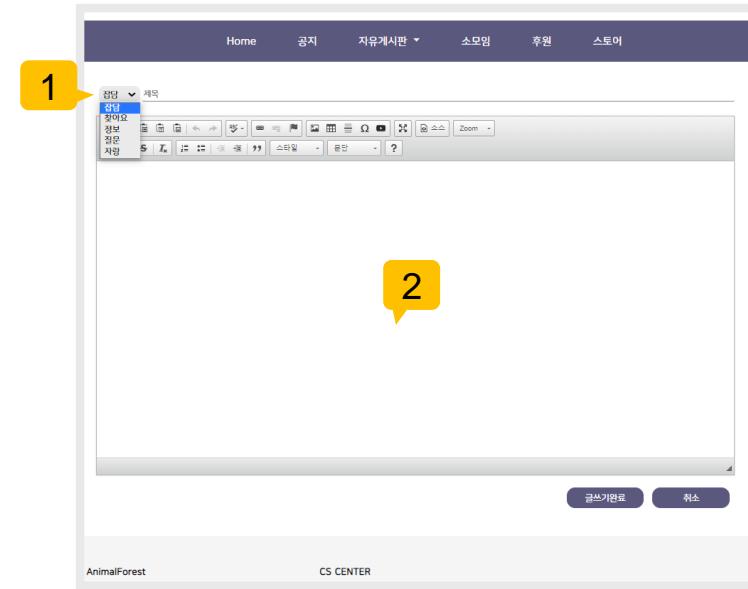
FbWrite.jsp

```

22 <input type="hidden" class="hdemail" value="${login.email}">
23
24 <% MemberDto user = (MemberDto)request.getSession().getAttribute("login");%>
25 <% FreeboardDto fb = (FreeboardDto)request.getAttribute("dto"); %>
26
27 <form method="post" name="fbwriteinfo" id="_fbwriteFrm" action="freeboard_boardwritedown.do">
28   <input type="hidden" name="email" value=<%user.getEmail()%>>
29   <input type="hidden" name="board_code" value="${board_code}">
30
31 <div class="container">
32
33   <div id="commentList">
34     <br><br>
35     <div>
36       <div>
37         <span><strong>${dto.category}</strong></span> <span id="cCnt"></span>
38       </div>
39       <div>
40         <table>
41           <tr>
42             <td> <label for="category"></label>
43             <select id="category" class="category" name="category" size="1" style="width:100px;"> 1
44               <option name="1" value="잡동">잡동</option>
45               <option name="2" value="찾아요">찾아요</option>
46               <option name="3" value="정보">정보</option>
47               <option name="4" value="질문">질문</option>
48               <option name="5" value="자랑">자랑</option>
49             </select>
50           </td>
51           <td> <input type="text" name="title" maxLength="40" class="title" placeholder="제목" style="width:1000px;">
52           </td>
53         </tr>
54         <tr>
55           <td colspan="2"><br>
56             <textarea type="text" name="content" class="content" placeholder="내용" id="p_content" rows="10" cols="50" ></textarea>
57           </td>
58         </tr>
59       </table>
60     </div>
61   </div>
62   <div>
63     <input type="hidden" id="b_code" name="b_code" value="" />
64   </div>
65 </div>
66 <br>
67 </div>
68
69 <div class="container" >
70   <input type="button" class="btnlist cancel" value="취소" onclick="location.href='freeboard_listview.do'">
71   <input type="button" class="btnlist ok" value="글쓰기완료" onclick="writefb()">
72   <br><br><br>
73 <!-- email -->
74 <input type="hidden" class="hdemail" value="${login.email}">
75
76 </div>
77 </form>
78

```

<Select태그> 활용 글 목록을 선택할 수 있는 옵션 구성



CKEditor. 리플레이스로 TextArea 대체 적용

```

84 <script>
85   function writefb(){
86     var form = document.fbwriteinfo;
87     $("#_fbwriteFrm").submit();
88   }
89 </script>
90
91 <script type="text/javascript"> 2
92   CKEDITOR.replace('p_content',
93     {filebrowserUploadUrl: '${pageContext.request.contextPath}/ckeditor/fileupload.do',
94      height: 500
95    });
96 </script>
97
98
99 </script> 3

```

3 이미지저장과정은 다음슬라이드 참고

작업 소스코드 – 자유게시판 글쓰기 구현(3/3)

글 작성 시, 이미지 저장 소스코드 및 적용

```

25 public class EditorController {
26     //에디터 브로우저에서 파일을 업로드하는 경우 동작
27     @RequestMapping(value = "fileupload.do", method=RequestMethod.POST)
28     @ResponseBody
29     public String fileUpload(HttpServletRequest req, HttpServletResponse resp,
30                             MultipartHttpServletRequest multiFile) throws Exception {
31         JsonObject json = new JsonObject();
32         PrintWriter printWriter = null;
33         OutputStream out = null;
34         MultipartFile file = multiFile.getFile("upload");
35         if(file != null){
36             if(file.getSize() > 0 && StringUtils.isNotBlank(file.getName())){
37                 if(file.getSize() > 10240000) {
38                     System.out.println("==업로드 허용용량 초과(10Mb) 재질검증==");
39                     return null;
40                 }
41                 else if(file.getContentType().toLowerCase().startsWith("image/")){
42                     try{
43                         String fileName = file.getName();
44                         byte[] bytes = file.getBytes();
45                         String uploadPath = req.getSession().getServletContext().getRealPath("/file");
46                         File uploadFile = new File(uploadPath);
47                         if(!uploadFile.exists()){
48                             uploadFile.mkdirs();
49                         }
50                         fileName = UUID.randomUUID().toString();
51                         uploadPath = uploadPath + "/" + fileName;
52                         out = new FileOutputStream(new File(uploadPath));
53                         out.write(bytes);
54
55                         printWriter = resp.getWriter();
56                         resp.setContentType("text/html");
57
58                         //이미지 파일 저장경로 지정
59                         String fileUrl = req.getContextPath() + "/file/" + fileName;
60                         // 파일 이름으로 등록
61                         // {"uploaded": 1, "fileName": "test.jpg", "url": "/img/test.jpg"}
62                         // 위의 경로로 리턴이 나가야함.
63                         json.addProperty("uploaded", 1);
64                         json.addProperty("fileName", fileName);
65                         json.addProperty("url", fileUrl);
66
67                         printWriter.println(json);
68                     }catch(IOException e){
69                         e.printStackTrace();
70                     }finally{
71                         if(out != null){
72                             out.close();
73                         }
74                         if(printWriter != null){
75                             printWriter.close();
76                         }
77                     }
78                 }
79             }
80         }
81         return null;
82     }
}

```

3-1

3-2

3-1

지정된 서버 폴더에 난수 적용 name으로 이미지 저장 준비.



3-1

서버 폴더 경로에 난수 네임으로 저장 적용.

이름	수정한 날짜	유형	크기
8f7a8ac7-c803-41df-90b3-bb3df36c4906	2020-07-15(수) 오후 3...	파일	1,176KB
52de75d8-2cee-472c-9d68-ed507a25d61	2020-08-02(일) 오후 3...	파일	20KB
96783fac-cd1c-41a4-b725-cf00cc95b6	2020-07-15(수) 오후 2...	파일	417KB
a09f4923-63cd-44d0-b2cc-f6aad686e8bc	2020-08-02(일) 오후 3...	파일	20KB
b4434b0e-0d5d-4604-875a-a294cdda8d68	2020-07-15(수) 오후 2...	파일	115KB
b1297045-f008-4312-832b-7dd8e9e6833f	2020-07-15(수) 오후 3...	파일	15KB
eba946d2-52c2-463b-823c-82e3a1d5d3b9	2020-07-15(수) 오후 2...	파일	128KB



작업 소스코드 – 자유게시판 댓글 구현(1/4)

댓글 작성 - 뷰 & 코드 설명

FreeboardController.java

```
// 게시글 본문상세
FreeboardDto fb = fbService.getdetail(seq);
model.addAttribute("dto", fb); // 오브젝트 dto

List cmlist = fbService.getcmlist2(likedto);
model.addAttribute("cmlist", cmlist); // 게시글에 달린 댓글 리스트
```

1 컨트롤러에서 전달받은 게시글 본문과, 댓글갯수에 따라 반복 등장하도록 구현

FbDetail.jsp

```
<tbody>
<%
for(int i = 0; i < cmList.size(); i++) {
    CommentDto cm = cmList.get(i);
    String from = cm.getRegdate();
    SimpleDateFormat transFormat = new SimpleDateFormat("yyyy-MM-dd HH:mm");
    Date to = transFormat.parse(from);
    String email = cm.getEmail();
}
%>
```



2 댓글 List 출력 시 댓글별 seq값이 <div id명>에 함께 적용되도록 구현

크롬 elements 창을 통한 확인

The screenshot shows a list of comments in a browser. One comment has a reply icon with the text 'Seq: 122' next to it. The browser's developer tools are open, specifically the Elements tab. A yellow box highlights the 'Seq: 122' text. Another yellow box highlights the 'color:#f5f5f5' style applied to the reply button's parent div. The DOM tree shows the reply button is enclosed in a div with the ID 'replycomment-click122'.



```
<td style="word-break: break-all; background-color:#f5f5f5">
<%
if(user == null || user.getEmail() == null){
    %>
        <font></font>
    <%
} else{
    if(cm.getEmail() != null & user.getEmail() != null){
        String commentId = cm.getEmail();
        String userId = user.getEmail();
        if(cm.getEmail().equals(user.getEmail())){
            %>
                <div id="reply-click<%=cm.getSeq()%>">
                    ,'<%=cm.getEmail()%>') style="cursor: pointer;">
                </div>
        }
    }
}
%>
<%>
<%
} else{ %>
    <font></font>
<%>
}
%>
</td>
```

작업 소스코드 – 자유게시판 댓글 구현(2/4)

댓글 순서 구현 상세설명

SEQ	BOARD_CODE	BOARD_SEQ	REF	STEP	DEPTH	EMAIL	CONTENT	LICKCOUNT	REPORTCOUNT	REGDATE	DEL
1	81 AFSTORY	81	15	0	0	jeffrey...	내가남긴댓글2	0	0	2020/07/20	0
2	82 AFSTORY	81	15	1	1	jeffrey...	답글123	0	0	2020/07/20	0
3	101 AFSTORY	101	16	0	0	jeffrey...	0001	0	0	2020/07/29	0
4	102 AFSTORY	101	17	0	0	jeffrey...	0002	0	0	2020/07/29	0
5	103 AFSTORY	101	18	0	0	jeffrey...	0003	0	0	2020/07/29	0
6	104 AFSTORY	101	16	3	1	jeffrey...	0001-01	0	0	2020/07/29	0
7	105 AFSTORY	101	16	1	1	jeffrey...	0001-02	0	0	2020/07/29	0
8	106 AFSTORY	101	16	4	2	jeffrey...	0001-01-01	0	0	2020/07/29	0
9	107 AFSTORY	101	16	2	2	jeffrey...	0001-02-01	0	0	2020/07/29	0
10	108 AFSTORY	101	16	5	3	jeffrey...	0001-01-01-01	0	0	2020/07/29	0

REF	STEP	DEPTH
16	0	0
16	1	1
16	2	2
16	3	1
16	4	2
16	5	3
17	0	0
18	0	0

댓글 및 대댓글 정렬 적용 방식

1 REF: 댓글 그룹을 구별하기 위한 값

2 STEP: 동일 그룹 내 댓글 순서를 정의하는 역할.

새로운 대댓글 작성 시 기준 댓글 STEP 보다 +1 증가시켜 적용.
또한, 동일한 REF 그룹의 댓글들이 적용값과 같거나 큰 경우
STEP+1씩 증가시켜 오름차순이 정상 적용되도록 구성.

3 DEPTH

0: 게시물에 직속형태로 달린 댓글

1: DEPTH가 0인 댓글에 답글을 단 경우

2: DEPTH가 1인 대댓글을 기준으로 답글을 단 경우

...



작업 소스코드 – 자유게시판 댓글 구현(3/4)

댓글 수정 소스 구현 설명

FbDetail.java

3

jQuery를 활용해 작업할 변수 위치를 next변수로 정의하고, .after 뒤에 수정입력폼이 나타날 수 있도록 구현

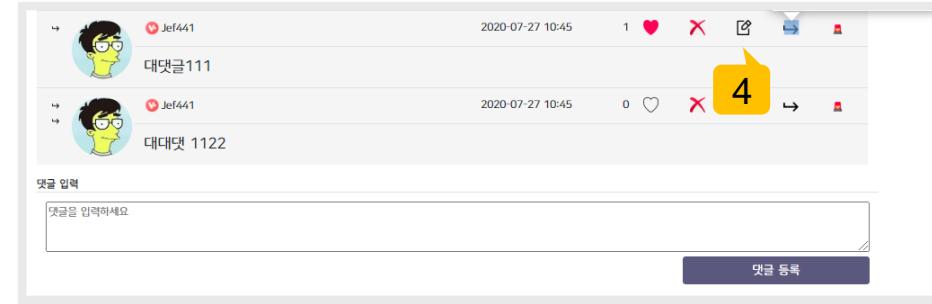
```
function updateOpenReply(th,seq,cmemail){
var email = $(".hdemail").val();
if(email == ""){
    alert("로그인이 필요한 페이지입니다.");
    location.href='login.do';
    return false;
}
if(cmemail != email){
    alert("작성자 본인만 수정/삭제할 수 있습니다.");
    return false;
}
if(count > 0){
    alert("글 수정과 답글은 동시에 실행할 수 없습니다");
    return false;
}
count++;

btn = $(th).parent().html()
btnlocation = $(th).parent().parent();
my = th;
next = $(my).parent().parent().parent().next().after();
mydiv = $(my).parent();
next2 = $(my).parent().parent().parent().next().after();
textbackup = next.text();

var content = $("#reply"+seq).html();
var backup = content

next.after("<tr id='updatereplyDiv"+seq+"><td colspan='9'><form><input type='text' value='"+content+"'/></form></td></tr>");
}
```

3



4

작성자가 '댓글 수정하기' 버튼을 누를 경우 표현되는 수정입력화면



화면출력을 위한
상세 코드는 중간생략

작업 소스코드 – 자유게시판 댓글 구현(4/4)

프로필 사진 / 좋아요 클릭여부 상태 적용

FbDetail.jsp

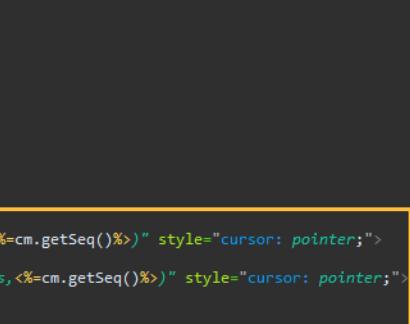
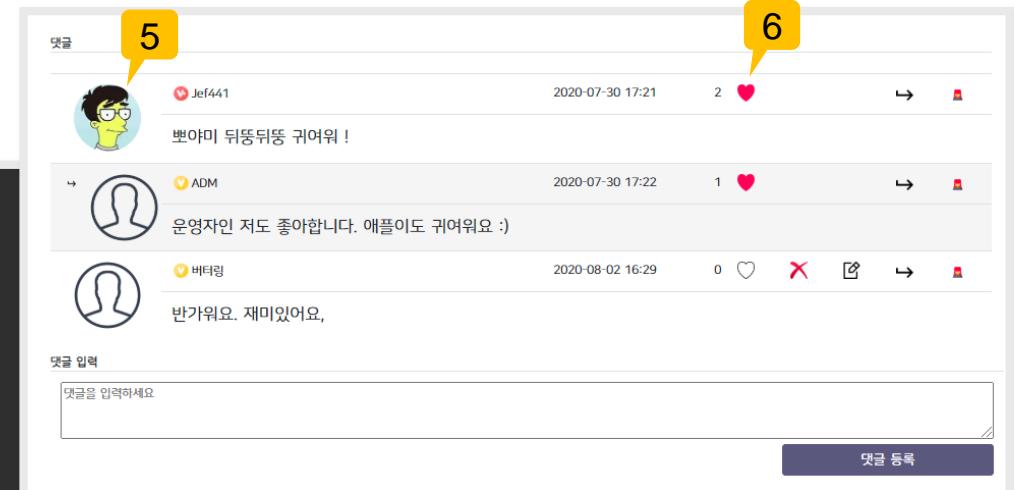
프로필사진 & 좋아요 -하트상태(♡||♡) 적용

```

<%if(cm.getDel() == 0){ %>
<tr>
    <%
    if(cm.getDepth() != 0){
    %>
        <td rowspan = "2" style="padding-top: 12px; padding-left: 0px; padding-right: 0px; padding-bottom: 0px; background-color:#f5f5f5">
        <%
        for(int j = 0; j < cm.getDepth(); j++) {
            %
        <img src='freeboard_resources/images/reply.png' width="10px" height="10px"
        style="padding-top: 0px; padding-left: 0px; padding-right: 0px; padding-bottom: 0px; margin-left:20px;"/>
        <%} %
        </td>
        <td rowspan = "2" style="padding-top: 12px; padding-left: 0px; padding-right: 0px; padding-bottom: 0px; background-color:#f5f5f5" >
        <%if(cm.getMyprofile_img() != null & cm.getMyprofile_img().contains("http")) {%
        <img src=<%=cm.getMyprofile_img()%> width="50px" class="profilephoto" height="50px">
        <%}else if(cm.getMyprofile_img() != null & cm.getMyprofile_img().contains('profileImg')){ %
        
        <%}else {%
        
        <%} %
    </td>
        <td style="word-break: break-all; background-color:#f5f5f5">
        <%=cm.getNickname()%>
    </td>
        <td style="word-break: break-all; background-color:#f5f5f5"><%=transFormat.format(to)%></td>
        <td style="word-break: break-all; background-color:#f5f5f5">
            <div class='cmlike' id='cmlikecount' display='inline'><%=cm.getLikecount()%></div>&ampnbsp&ampnbsp&ampnbsp;
            <%if(cm.getViewer_emailcheck() >= 1) {%
            " width="20px" height="20px" onclick="commentlikecheck(this,<%=cm.getSeq()%>)" style="cursor: pointer;"/>
            <%}else {%
            " width="20px" height="20px" onclick="commentlikecheck(this,<%=cm.getSeq()%>)" style="cursor: pointer;"/>
            <%} %
        </td>
    </tr>

```

6



```

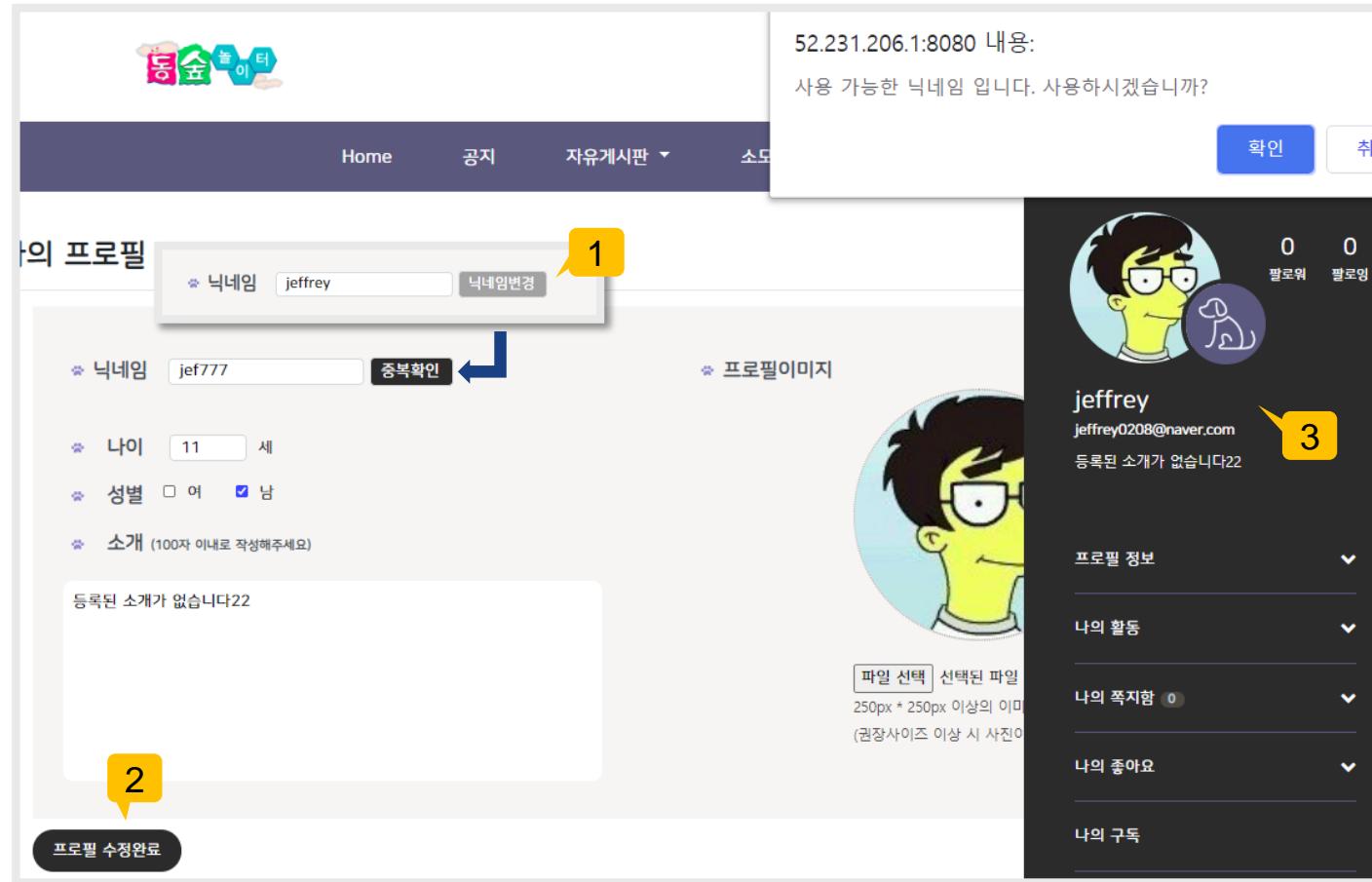
public class CommentDto implements Serializable{
    private int seq;
    private String board_code;
    private int board_seq;
    private int ref;
    private int step;
    private int depth;
    private String email;
    private String nickname;
    private String myprofile_img;
    private String filename;
    private String content;
    private int likecount;
    private int reportcount;
    private String regdate;
    private int del;
    private int viewer_emailcheck;
    // 코멘트에 있는 좋아요를 출석체크하는 경우 - 1:라이크 , 0:디스라이크
}

```

Cf. CommentDto.java

작업 소스코드 – MyPage 프로필 수정(1/2)

MyPage 정보수정 흐름 설명



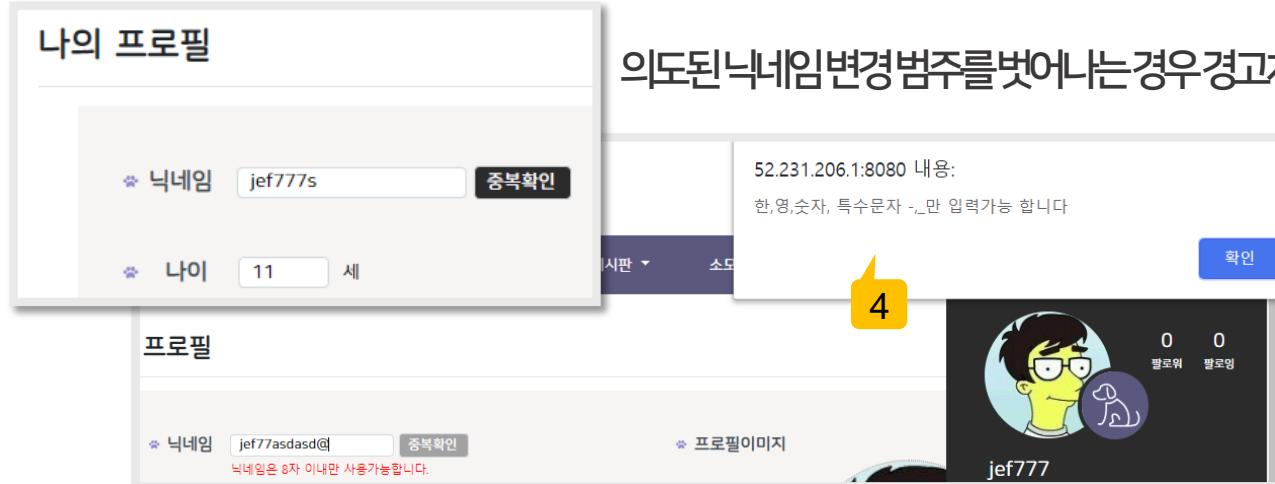
■ MyPage 정보수정 로직

- 1 **닉네임변경의 경우 중복확인 과정 있음.**
- 2 **그 외 프로필사항은 수정사항 기입 후 좌측 하단(프로필 수정완료) 버튼으로 적용**
- 3 **수정완료 후 세션에 저장된 (구) 유저정보를 DB로부터 다시 불러온 (신) 유저정보로 업데이트**



작업 소스코드 – MyPage 프로필 수정(2/2)

의도하지 않은 MyPage 정보수정 사전 경고



MyProfile.jsp - View 영역

```
<div class="mypageHomebox_1">
  <div class="input_row">
    <p class="input_title">닉네임</p>
    <span>
      <input name="nickname" type="text" size="20px" value="${login.nickname }" readonly="readonly">
      <input type="button" id="nickChange_btn" class="nickchange" value="닉네임변경">
      <input type="button" id="nickChk_btn" class="nickchange displayis" value="사용하기">
    </span>
    <p><font class="nickChgMsg"></font></p>
  </div>
</div>
```

5

MyProfile.jsp - js 영역

```
// keyup 이벤트
$('input[name=nickname]').on('keyup', function(event) {
  // 표현식 제한
  4 var regexp = /^[ㄱ-ㅎㅏ-ㅣㅏ-ㅣㄱ-ㅎa-zA-Z0-9\-\_]/gi;
  var nameval = $(this).val();
  if( regexp.test( nameval ) ){
    alert("한, 영, 숫자, 특수문자 -, _만 입력 가능 합니다.");
    $(this).val(nameval.replace(regexp, ''));

    // 글자수제한
    if($(this).val().length > 8) {
      $(".nickChgMsg").text("닉네임은 8자 이내만 사용 가능합니다.");
      $(".nickChgMsg").attr("color", "red");
      $("#nickChk_btn").prop("disabled", true);
      $("#nickChk_btn").css("background-color", "#a2a2a2");
      //
    }
    if($(this).val().length <= 8) {
      $(".nickChgMsg").text("");
      $("#nickChk_btn").prop("disabled", false);
      $("#nickChk_btn").css("background-color", "#2b2b2b");
    }
  });
});
```

감사합니다.

