

포트폴리오



정현희

깃헙 : <https://github.com/hyunniesss>

블로그 : <https://blog.naver.com/wjdugusgml997>

이메일 : wjdugusgml997@naver.com

소개

하면 된다! 끈기와 열정을 가진 개발자, 정현희

안녕하세요. 신입 개발자 정현희입니다.

졸업 직후 SSAFY 3기에 입과하여 프론트엔드 개발부터 백엔드 개발까지 웹개발자로서의 초석을 다졌습니다.

함께 일하게 될 개발자님들과의 코드 리뷰를 통해 성장할 미래를 기대하고 있습니다.

개인 블로그에서 알고리즘 위주의 문제풀이를 공유하고 있습니다.

보유 기술

언어

- JavaScript
- Java

프레임워크

- Spring framework & Spring boot
- Vue.js

데이터베이스

- mySQL
- mariaDB
- Redis

협업 도구

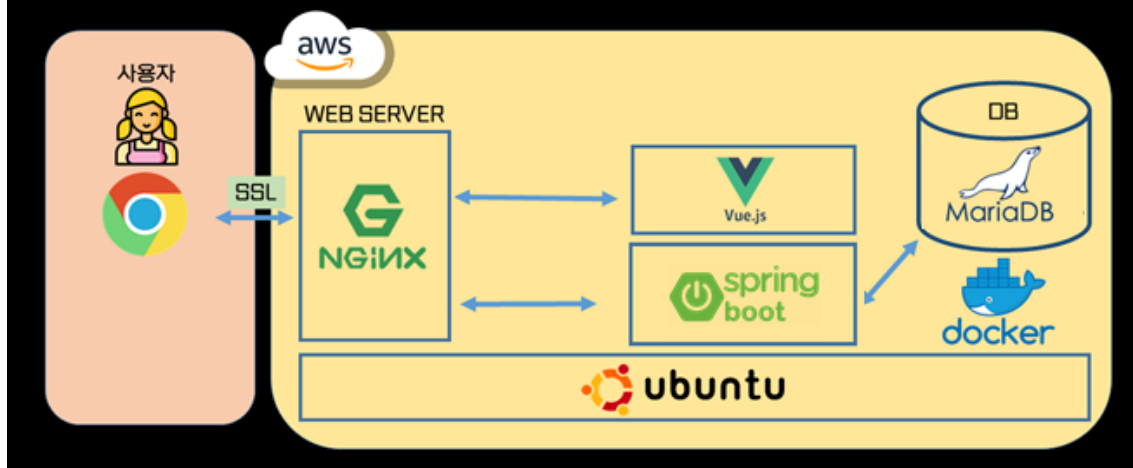
- Git
 - git flow
- JIRA

프로젝트

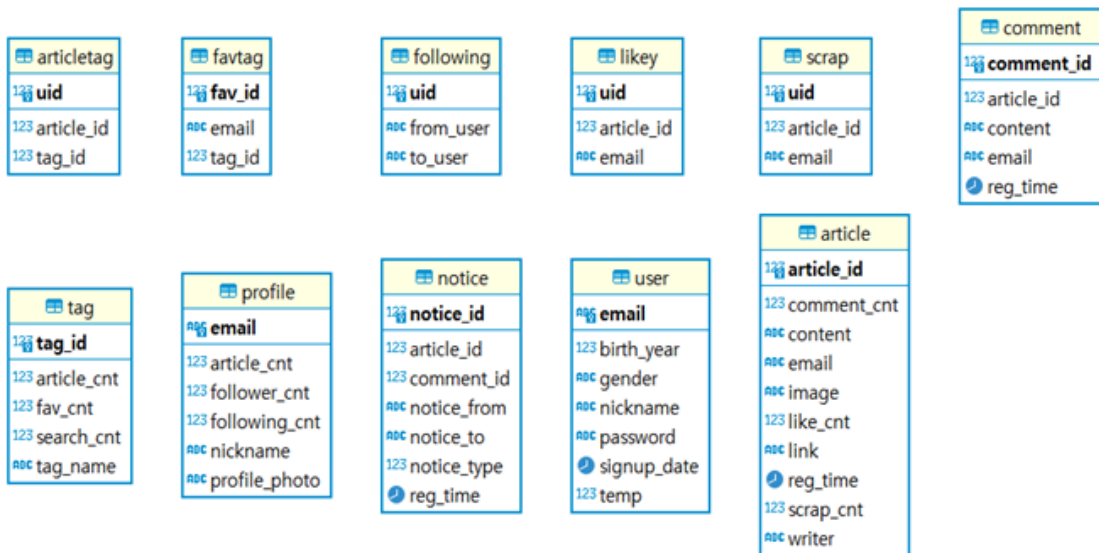
1. Tugether

- 개발 기간 : 2020.07.07 ~ 2020.08.21 (7 weeks)
- 설명 : 유튜브 취향을 공유할 수 있는 SNS 서비스
- 사용 기술 : spring boot, jpa, lombok, mariaDB
- 역할(백엔드 개발)
 - 무한 스크롤을 적용한 메인피드 조회
 - 검색 조건(ex. 태그 또는 닉네임으로 검색)에 맞춰 DB에서 데이터 조회
 - 댓글 기능
 - 팔로우 기능
 - 알림 기능
- 시스템 구성도

SYSTEM ARCHITECTURE



- ERD



- 소스 코드 : https://github.com/hyunniesss/PJT1_Together
- UCC 영상 : <https://youtu.be/OqeJrNeo9rA>
- 소스 코드 패키지 구조 및 소스 코드 일부

```

> src/main/java
> com.web.curation
> com.web.curation.config
> com.web.curation.controller
> com.web.curation.dto
> com.web.curation.dto.account
> com.web.curation.dto.article
> com.web.curation.dto.comment
> com.web.curation.dto.follow
> com.web.curation.dto.notice
> com.web.curation.dto.profile
> com.web.curation.dto.tag
> com.web.curation.jwt.exception
> com.web.curation.jwt.interceptor
> com.web.curation.jwt.service
> com.web.curation.repo
> com.web.curation.service.account
> com.web.curation.service.articlewrite
> com.web.curation.service.comment
> com.web.curation.service.feed
> com.web.curation.service.follow
> com.web.curation.service.likey
> com.web.curation.service.mypage
> com.web.curation.service.notice
> com.web.curation.service.profile
> com.web.curation.service.scrap
> com.web.curation.service.search
> com.web.curation.service.stats
> com.web.curation.service.tag

```

```

/*
 * 1. 태그기반 피드
 * front에서 email get -> favtag 테이블에서 해당 이메일로 tag_id 찾아와
 * ArticleTag 테이블에서 tag_id로 article_id를 리스트로 받아와
 * Article 테이블에서 article_id로 article 리스트 다 가져와
 * ArticleTag 테이블에서 article_id로 List<tag_id> 가져와 => Tag 테이블에서 tag_id로 tag_name 가져와
 *
 * 2. 팔로우기반 피드
 * following 테이블에서 from_user=email로 to_user 리스트 찾아와
 * article 테이블에서 uid = email인 List<article>로 다 가져와
 */

// 태그기반 또는 팔로우기반 피드 조회
@GetMapping("/mainfeed/fromto")
public List<FrontArticle> getAfterMainFeed(@RequestParam boolean tag, @RequestParam int from, @RequestParam int to,
    HttpServletRequest request) {
    String email = ((Map<String, Object>) jwtService.getDecodeToken(request.getHeader("jwt-auth-token"))).getBody()
        .get("AuthenticationResponse").get("email").toString();
    List<FrontArticle> result = null;
    if (tag) {
        result = feedService.findArticleListByTag(email, from, to);
    } else {
        result = feedService.findArticleListByFollow(email, from, to);
    }
    return result;
}

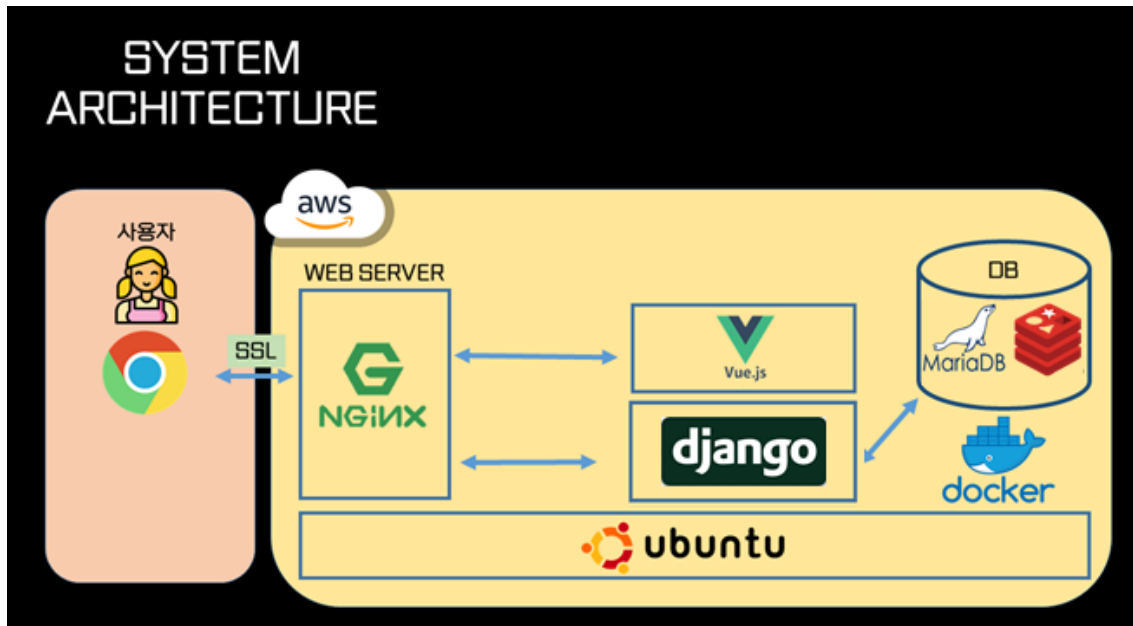
// 팔로우스크롤용 메인피드 조회
@GetMapping("/mainfeed")
public List<FrontArticle> getMainFeed(@RequestParam boolean tag, @RequestParam int limit,
    HttpServletRequest request) {
    String email = ((Map<String, Object>) jwtService.getDecodeToken(request.getHeader("jwt-auth-token"))).getBody()
        .get("AuthenticationResponse").get("email").toString();
    int pageNum = Integer.parseInt(request.getHeader("pageNum"));
    System.out.println("pageNum: " + pageNum);
    PageRequest pageRequest = PageRequest.of(limit, 2);
    List<FrontArticle> result;

    if (tag)
        result = feedService.findByPageRequestTag(pageRequest, email);
    else
        result = feedService.findByPageRequestFollow(pageRequest, email);
}

```

2. Meomullae

- 개발 기간 : 2020.08.31 ~ 2020.10.08 (6 weeks)
- 설명 : 빅데이터 기반 주변 음식점 추천 서비스
- 사용 기술 : Vue.js, Vuetify, axios
- 역할(프론트엔드 개발)
 - vuetify를 활용한 프론트 페이지의 전반적인 디자인
 - 로그인 및 회원가입 페이지
 - daum API를 이용한 주소 찾기 기능
- 시스템 구성도



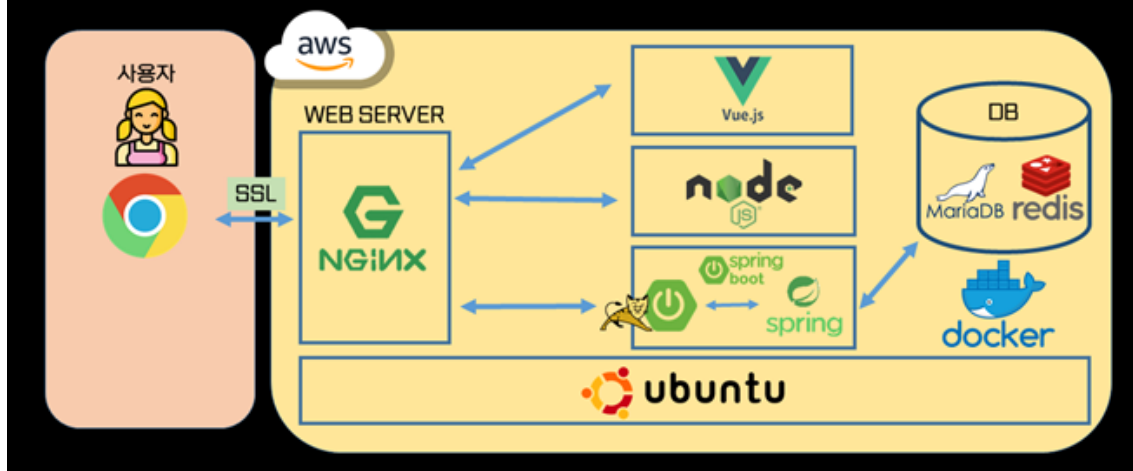
- 소스 코드 : https://github.com/hyunniesss/PJT2_Meomullae
- UCC 영상 : <https://youtu.be/D3KPND9Nt7c>
- 화면 예시



3. 마이리틀댕댕

- 개발 기간 : 2020.10.12 ~ 2020.11.27 (7 weeks)
- 설명 : 유기견 입양을 권장하고 입양의 허들을 낮출 수 있는 웹사이트
- 사용 기술 : spring boot, spring security, JWT 및 JWT Filter, coolSMS, redis, mariaDB
- 역할(백엔드 및 프론트엔드 개발)
 - 백엔드 - Spring security 와 JWT Filter를 사용한 사용자 인증 및 인가
 - 백엔드 - Redis를 세션 서버로 적용
 - 백엔드 - 영상 업로드 기능
 - 프론트엔드 및 백엔드 - 입양 신청서 작성 및 조회, 수정, 삭제
 - 백엔드 - coolSMS를 이용한 문자 기능
- 시스템 구성도

SYSTEM ARCHITECTURE



- 소스 코드 : https://github.com/hyunniesss/PJT3_MyLittleDaengDaeng
- UCC 영상 : <https://youtu.be/2DJOlPAY0xo>
- 소스 코드 패키지 구조 및 소스 코드 일부

```

src/main/java
├── com.daeng.nyang
├── com.daeng.nyang.config
├── com.daeng.nyang.controller
├── com.daeng.nyang.dto
├── com.daeng.nyang.jwt
├── com.daeng.nyang.repo
├── com.daeng.nyang.service
├── com.daeng.nyang.service.animal
├── com.daeng.nyang.service.changepassword
├── com.daeng.nyang.service.email
├── com.daeng.nyang.service.findid
├── com.daeng.nyang.service.findpw
├── com.daeng.nyang.service.openapi
├── com.daeng.nyang.service.survey
├── com.daeng.nyang.service.user
├── com.daeng.nyang.service.video

```

```

@Configuration
@EnableWebSecurity
@EnableGlobalMethodSecurity(prePostEnabled = true)
public class SecurityConfig extends WebSecurityConfigurerAdapter {

    @Autowired
    private JwtRequestFilter jwtRequestFilter;

    @Autowired
    private JwtAuthenticationEntryPoint jwtAuthenticationEntryPoint;

    final String corsOrigin = "**";

    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http
            .addFilterBefore(jwtRequestFilter, UsernamePasswordAuthenticationFilter.class)
            // security에서 제공하는 UsernamePasswordAuthenticationFilter 이전에 내가 만든 JwtRequestFilter 먼저 처리해 줌
            .httpBasic().disable() // 기본적으로 제공하는 페이지 비활성화
            .csrf().disable() // 요청주소 방지 비활성화
            .cors().and(). // 아래 만든 CorsConfigurationSource로 cors 이용
            .formLogin().disable(). // security에서 제공하는 formLogin 비활성화
            .sessionManagement().sessionCreationPolicy(SessionCreationPolicy.STATELESS). // 세션관리와 세션 생성 X 사용 X
            and().authorizeRequests(). // 요청 시
            requestMatchers(CorsUtils::isPreFlightRequest).permitAll(). // CORS 허가 후
            antMatchers("/newuser/login", "/newuser/signup").anonymous(). // /newuser/login, /newuser/signup 으로 시작하는 url 요청 시 인증필요 없이 컨트롤러 진입
            antMatchers("/user").hasAnyRole("ADMIN", "USER"). // /user 로 시작하는 url 요청 시 ADMIN, USER 중 하나의 권한을 가지고 있어야 컨트롤러 진입 가능
            antMatchers("/admin").hasRole("ADMIN"). // /admin 으로 시작하는 url 요청 시 ADMIN 권한을 가진 권한이 컨트롤러의 진입 가능
            antMatchers("/newuser/**").permitAll(). // /newuser/ 로 시작하는 모든 요청은 인증이 안되도 컨트롤러의 진입 가능
            antMatchers("/newuser").permitAll().
            and().authorizeRequests(). // 그 외
            anyRequest(). // 어떤 요청이든
            authenticated(). // 인증된 사용자가 접근 허용
            and().exceptionHandling().authenticationEntryPoint(jwtAuthenticationEntryPoint); // exception 발생 시 내가 만든 JwtAuthenticationEntryPoint로 exception 전달하여 처리
    }
}

```

- 화면 예시

[illegible]



한울림
Haeil

이름: _____

학번: _____

4기출 문제

1. 일출시간 _____

2. 일출각도 _____

3. 일출지 _____

4. 일출 _____

5. 일출 _____

3월 21일 00:00:00



일출문제 2차

1. 일출시간: 06:00:00 _____

2. 일출각도: 30° _____

3. 일출지: 서울 _____

4. 일출: 서울 _____

5. 일출: 서울 _____

일출문제 1차

1. 일출시간: 06:00:00 _____

2. 일출각도: 30° _____

3. 일출지: 서울 _____

4. 일출: 서울 _____

5. 일출: 서울 _____

| 인사관리 | | | | 인사 |
|------|--------|--------|--------|----|
| 인사관리 | | | | 인사 |
| 구분 | 항목 | 내용 | 비고 | |
| 1 | 인사관리제도 | 인사관리제도 | 인사관리제도 | |
| 2 | 인사관리제도 | 인사관리제도 | 인사관리제도 | |
| 3 | 인사관리제도 | 인사관리제도 | 인사관리제도 | |
| 4 | 인사관리제도 | 인사관리제도 | 인사관리제도 | |
| 5 | 인사관리제도 | 인사관리제도 | 인사관리제도 | |