



ElastiQuer (정재윤)

개요 [↗](#)

프로젝트 목적 및 간단 설명

- 기술 키워드로 오픈 데이터를 쉽고 빠르게 검색하고, 최신 트렌드를 한눈에 분석할 수 있는 플랫폼입니다.
- 사용자 맞춤형 데이터 추천과 시각화 분석을 통해 혁신적인 인사이트를 제공합니다.

주요기능구현 [↗](#)

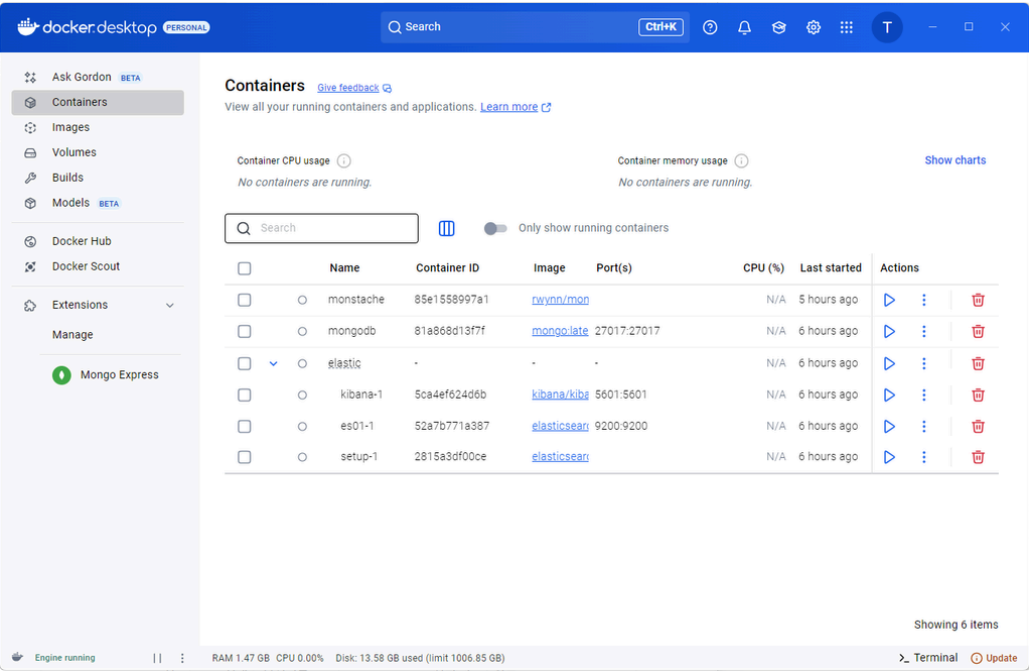
주요 기능 구현 [↗](#)

기능명	설명	설정	완료여부
Docker 서버 구축 및 환경 설정	Elastic, Kibana, MongoDB, Monstache 환경 세팅	Ubuntu wsl2, Docker Compose, Docker Desktop, Docker yml, Elastic Search, Kibana, MongoDB, Monstache	완료
Virtual Box - Ubuntu Linux 서버 구축	Virtual Box를 이용한 Ubuntu Linux 서버 환경 구축	Virtual Box 설치 및 Ubuntu 이미지 설정	완료
Spring Security, Jwt 적용	Spring Security와 JWT를 이용한 인증 및 권한 관리 구현	Spring Boot 프로젝트 설정	완료
Elastic Search Nori 형태소 적용	Nori 형태소 분석기를 Elastic Search에 적용	Elastic Search 설정 파일 수정	완료
Elastic Search 구현	Elastic Search를 통한 데이터 검색 기능 구현	Elastic Search 인덱스 및 매핑 설정	완료
Favorites 구현	사용자가 즐겨찾는 항목을 저장하고 관리하는 기능 구현	데이터베이스 모델링 및 API 개발	완료
HackerNews API 구현	HackerNews API를 이용한 데이터 수집 및 표시 기능 구현	API 호출 및 데이터 파싱	완료
MongoDB Spring Boot CRUD 구현	MongoDB와 Spring Boot를 이용한 CRUD 기능 구현	MongoDB 설정 및 Repository 생성	완료
StackOverflow 구현	StackOverflow API를 이용한 질문 및 답변 표시 기능 구현	API 호출 및 데이터 처리	완료

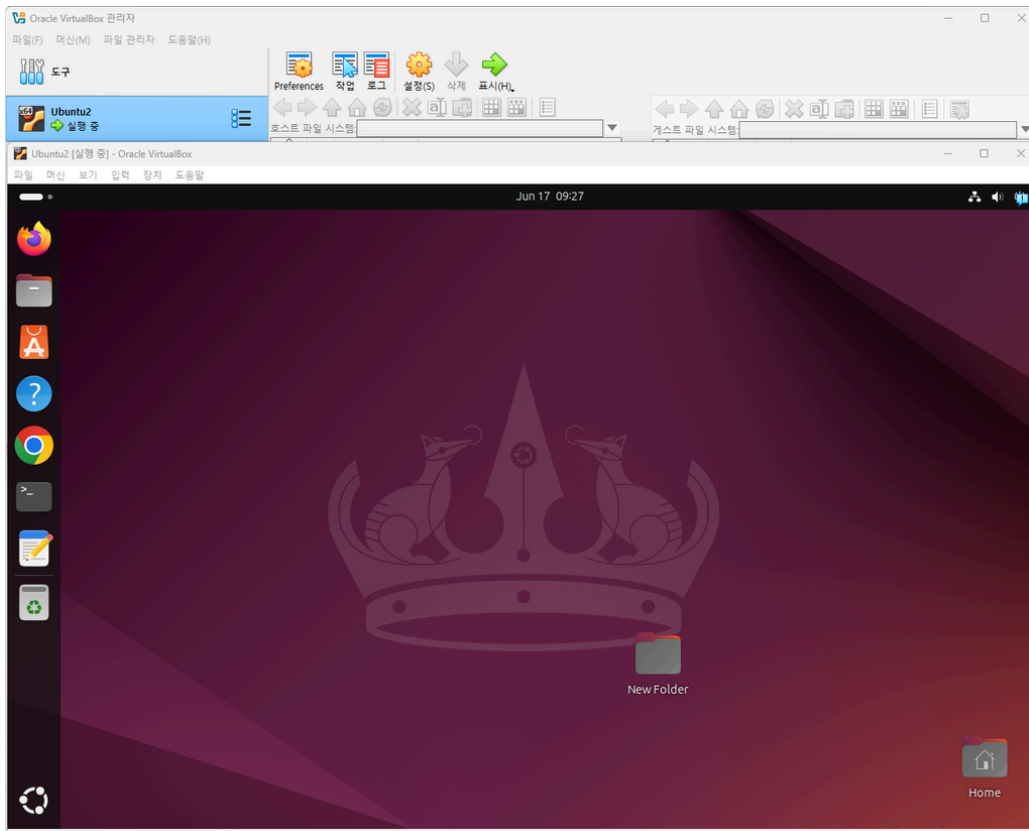
User CRUD 적용	사용자 정보를 관리하는 CRUD 기능 구현	사용자 모델 및 API 개발	완료
Python Flask 서버 구축	Flask를 이용한 간단한 웹 서버 구축	Flask 설치 및 기본 라우팅 설정	완료
Hanspell 및 Google Trends 연결	Hanspell API와 Google Trends API를 연결하여 데이터 처리	API 호출 및 결과 표시	완료
React 요소 일부 적용	React를 이용한 UI 요소 수정 및 이벤트 핸들링 구현	React 컴포넌트 수정	완료
Elastic Search 검색 결과 필터링, 하이라이트, 정렬 구현	Elastic Search의 검색 결과를 필터링하고 하이라이트 및 정렬 기능 추가	쿼리 수정 및 결과 처리	완료

이미지 및 시각자료

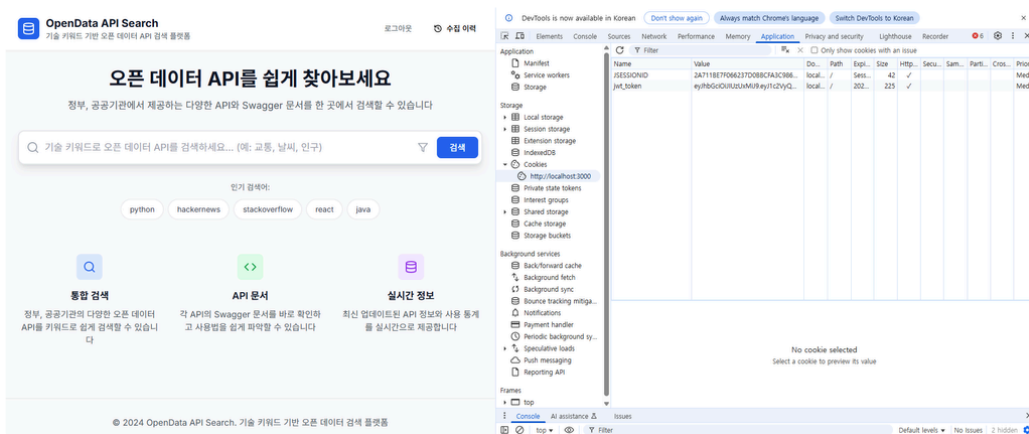
Docker 서버 구축



Virtual box 환경 세팅



JWT 적용



Nori 형태소 분석 적용

```
PUT /mydb.stackoverflowquestions
{
  "settings": {
    "analysis": {
      "analyzer": {
        "nori": {
          "tokenizer": "nori_tokenizer"
        }
      }
    }
  },
}
```

elastic search 검색 결과 및 즐겨찾기 기능

iterator and iterable classes in python

Why do we return self in the iter method when we define the next method in the iterable and iterator classes?
This topic was taught in the course, but it was hard to understand and I didn't understand why we had to return self.

class PowTwo:...

python iterator iterable

API 문서 즐겨찾기 사이트 방문

데이터셋	상태	수집 시간	레코드 수	파일 크기	액션
iterator and iterable classes in python	성공	2025-06-16 19:45:15	12,345 건	23	👁️ ⬇️ 🗑️
Exception Value:failed to find libmagic. Check your installation in windows 7	성공	2025-06-16 09:27:11	12,345 건	23	👁️ ⬇️ 🗑️
How can I use Python packages installed with apt-get in a Python interpreter installed with pyenv?	성공	2025-06-13 18:24:31	12,345 건	23	👁️ ⬇️ 🗑️

4. 코드 및 코드 설명 [🔗](#)

4.1 주요 코드 예시 [🔗](#)

4.1.1 주요 코드 [🔗](#)

```

1 @PostConstruct
2 public void init() {
3     RestClient restClient = RestClient.builder(
4         new HttpHost(host, port, "https")
5     ).setHttpClientConfigCallback(httpClientBuilder ->
6         httpClientBuilder.setDefaultHeaders(
7             Collections.singletonList(
8                 new BasicHeader("Authorization", "ApiKey " + apiKey)
9             )
10        )
11    ).build();
12
13    ElasticsearchTransport transport = new RestClientTransport(restClient, new JacksonJsonpMapper());
14
15    this.client = new ElasticsearchClient(transport);
16 }

```

4.1.2 코드 설명 [🔗](#)

- Elasticsearch 서버와 통신할 수 있는 클라이언트 객체 준비
 - PostConstruct를 사용한 자동 초기화
 - https 프로토콜을 명시해 보안 연결 사용
 - 기존 인증보다 안전성 높은 API Key 인증
-

4.1.3 주요 코드 [🔗](#)

```
1 public String translate(String text) throws Exception {
2     WebClient webClient = WebClient.create(base);
3
4     String response_translate = webClient.post()
5         .uri("/translate")
6         .contentType(MediaType.APPLICATION_FORM_URLENCODED)
7         .bodyValue("auth_key=" + apiKey + "&text=" + text + "&target_lang=EN")
8         .retrieve()
9         .bodyToMono(String.class)
10        .block();
11    String translatedText = extractTranslatedText( response_translate);
12    return translatedText;
13
14 }
```

4.1.4 코드 설명 [🔗](#)

- DeepL API연동 텍스트 번역 기능
 - 비동기 HTTP 클라이언트 WebClient 사용
-

4.1.5 주요 코드 [🔗](#)

```
1 SearchRequest request = SearchRequest.of(s -> s
2     .index(indexName) // 정확한 인덱스로 고정
3     .from(from) // 'from' 값으로 페이징 시작 위치 지정
4     .size(20)
5     .query(q -> {
6         if (searchQuery == null || searchQuery.isEmpty()) {
7             log.info("elastic controller : " + searchQuery);
8             return q.matchAll(m -> m);
9         } else {
10            log.info("elastic controller2 : " + searchQuery);
11            return q.bool(b -> b
12                .should(m -> m.multiMatch(mm -> mm
13                    .fields("title^5", "body^2", "tags")
14                    .query(searchQuery)
15                    .type(TextQueryType.BestFields)
16                    .fuzziness("AUTO")
17                    .operator(Operator.Or)
18                    .minimumShouldMatch("70%")
19                ))
20            // 카테고리 필터링 추가 - 검색 정확도는 높아지지만 데이터셋 부족으로 사용 x
21            .filter(f -> {
22                if (searchCategory != null && !searchCategory.isEmpty()) {
23
24                }
```

```

25 //                                return f.term(t -> t.field("tag").value(searchCategory));
26 //                                } else {
27 //                                return null; // 필터 없음
28 //                                }
29 //                                })
30 //                                );
31 //                                }
32 //                                })
33 // 하이라이트 <em> 태그 붙여서 나옴
34 .highlight(h -> h
35     .fields("title", f -> f)
36     .fields("body", f -> f)
37     .fields("tags", f -> f)
38 )
39
40 // 정렬 조건
41 .sort(sortBuilder -> {
42     if ("date".equals(sortBy)) {
43         return sortBuilder.field(f -> f.field("score").order(SortOrder.Desc));
44     } else if ("popularity".equals(sortBy)) {
45         return sortBuilder.field(f -> f.field("view_count").order(SortOrder.Desc));
46     } else if ("relevance".equals(sortBy)) {
47         return sortBuilder.field(f -> f.field("score").order(SortOrder.Desc));
48     } else {
49         // 기본 정렬 조건 (예: _score 내림차순)
50         return sortBuilder.field(f -> f.field("_score").order(SortOrder.Desc));
51     }
52 })
53 );

```

4.1.6 코드 설명 [🔗](#)

- Elasticsearch 조건 검색
- 검색쿼리, 페이징, 필터링, 하이라이트, 정렬
- 검색어가 있을 경우에는 유사도 기반으로 없을 경우에는 전체 문서를 조회

4.1.7 주요 코드 [🔗](#)

```

1 // JWT 기반 사용자 정보 반환
2 @GetMapping("/user/me")
3 public ResponseEntity<?> getCurrentUser(Authentication authentication) {
4     // 인증이 없거나, 인증이 anonymousUser이거나, PrincipalDetails가 아니면 401 반환
5     if (authentication == null || !authentication.isAuthenticated() ||
6         authentication.getPrincipal() == null ||
7         "anonymousUser".equals(authentication.getPrincipal()) ||
8         !(authentication.getPrincipal() instanceof PrincipalDetails)) {
9         return ResponseEntity.status(401).body("Unauthorized");
10    }
11    PrincipalDetails principal = (PrincipalDetails) authentication.getPrincipal();
12    UserDTO user = principal.getUser();
13    // DB에서 최신 사용자 정보 조회
14    HashMap<String, String> param = new HashMap<>();
15    param.put("userId", user.getUserId());
16    UserDTO fullUser = userService.getUserInfo(param);
17    return ResponseEntity.ok(fullUser);
18 }

```

4.1.8 코드 설명 [🔗](#)

- JWT 인증 기반 사용자의 최신 정보 반환 API
 - JWT 인증을 통한 로그인 사용자 조회 - 보안 강화
 - 세션에 저장된 사용자 정보가 아닌 DB에서 최신 정보 조회 - 실시간 정보
 - PrincipalDetails 커스텀 인증 객체를 활용 - 확장성
-

4.1.7 주요 코드 [🔗](#)

```
1 // JWT 기반 사용자 정보 반환
2 @GetMapping("/user/me")
3 public ResponseEntity<?> getCurrentUser(Authentication authentication) {
4     // 인증이 없거나, 인증이 anonymousUser이거나, PrincipalDetails가 아니면 401 반환
5     if (authentication == null || !authentication.isAuthenticated() ||
6         authentication.getPrincipal() == null ||
7         "anonymousUser".equals(authentication.getPrincipal()) ||
8         !(authentication.getPrincipal() instanceof PrincipalDetails)) {
9         return ResponseEntity.status(401).body("Unauthorized");
10    }
11    PrincipalDetails principal = (PrincipalDetails) authentication.getPrincipal();
12    UserDTO user = principal.getUser();
13    // DB에서 최신 사용자 정보 조회
14    HashMap<String, String> param = new HashMap<>();
15    param.put("userId", user.getUserId());
16    UserDTO fullUser = userService.getUserInfo(param);
17    return ResponseEntity.ok(fullUser);
18 }
```

4.1.8 코드 설명 [🔗](#)

- JWT 인증 기반 사용자의 최신 정보 반환 API
 - JWT 인증을 통한 로그인 사용자 조회 - 보안 강화
 - 세션에 저장된 사용자 정보가 아닌 DB에서 최신 정보 조회 - 실시간 정보
 - PrincipalDetails 커스텀 인증 객체를 활용 - 확장성
-

4.1.9 주요 코드 [🔗](#)

```
1 from flask import Flask, request, jsonify
2 from hanspell import spell_checker
3
4 app = Flask(__name__)
5
6 @app.route('/correct')
7 def correct():
8     text = request.args.get('text', '')
9     try:
10        result = spell_checker.check(text)
11        return jsonify({'corrected': result.checked})
12    except Exception as e:
13        return jsonify({'error': str(e)}), 500
14
15 if __name__ == '__main__':
16    app.run(port=5001)
```

4.1.10 코드 설명 [🔗](#)

- 한글 맞춤법 검사 API
 - hanspell 라이브러리 활용
 - 한글 맞춤법 자동화, API 서비스화, 예러 핸들링
-

이슈 [🔗](#)

- ☒ AWS 서버 메모리 이슈 - VirtualBox Ubuntu linux 서버 이전
- ☒ Elasticsearch 연동 및 인증 - 보안 연결 설정 (https), keytool
- ☐