




빠르게 코드가 생성되는 시대에, 설명할 수 있는 작업물을 개발하겠습니다.


SKILLS


Java 
- Java 버전에 따른 기능 차이를 이해하고 있음
Star 5000++의 오픈소스 Armeria기여 경험

Spring 
- Spring Boot로 REST API 서버 개발 가능
- Spring Security로 JWT인증 적용 가능
- Spring AI를 활용하여 RAG 구현 가능

MySQL 
- InnoDB에 의한 인덱싱을 알고있음
- 실행계획을 통해 쿼리가 어떻게 실행될지 예측 가능

JPA 
- N+1문제 해결 및 영속성 컨텍스트 이해

MyBatis 
- 동적 쿼리(if/foreach)를 활용한 조건별 조회 구현
resultMap 기반 객체 매핑

Vue.js 
- Pinia 상태관리 적용
- Vue Router를 통한 페이지 흐름 및 인증 제어
- Axios기반 API연동 경험

CERTIFICATES / Languages

2025.12 SQLD
2025.12 정보처리기사
2024.05 TOPCIT(수준3)

2024.10 TOEIC (880)
2024.09 TOEIC Speaking (AL 160)

EDUCATION / EXPERIENCE

(2025.07 ~ 2026.06) 삼성청년SW·AI아카데미 14기 재학중
- Java, Spring Boot, MyBatis, Vue.js
(2019 ~ 2025.02) 중앙대학교 소프트웨어학과 졸업(우등상)

PROJECT



(2026.04 ~ 2026.05) GitHub

Github repository 코드 분석 서비스

3p - 9p

pammap

종이 팜플릿을 디지털화한 실시간 축제 정보 안내 서비스

(2026.01 ~ 2026.02) PAMMAP

지역축제 정보 제공 서비스

10p - 13p



(2025.12 ~ 2025.12) Nesto

부동산 청약정보 제공 서비스

14p - 19p



(2025.08 ~ 2026.03) Armeria 오픈소스 기여

API 문서화 서비스에 상속, 다형성 정보를 반영하도록 PR제출

20p - 25p

GitHub 저장소 분석 서비스 GitHubble

2026.04 ~ 2026.05 (인원 : 6명)

프로젝트 개요

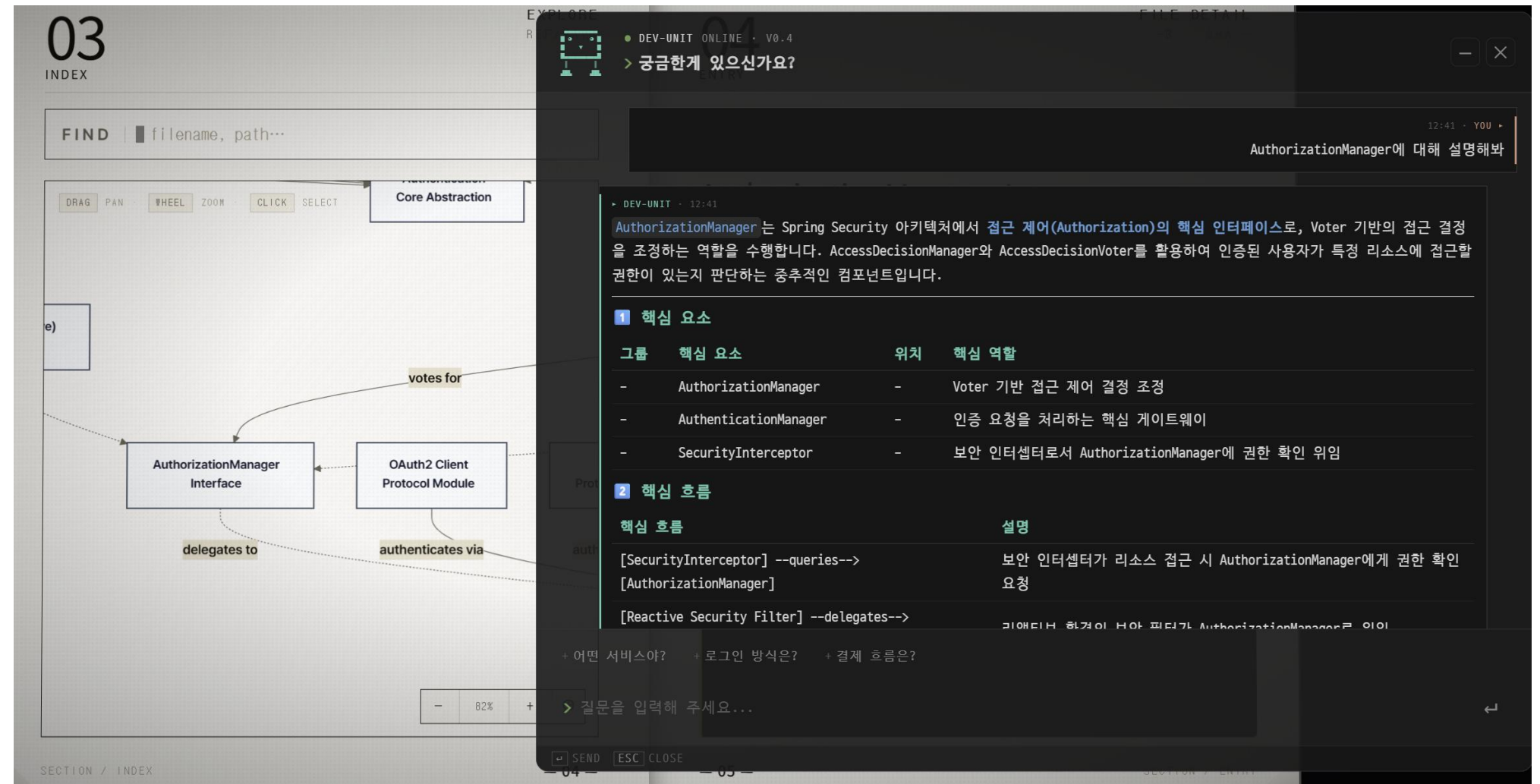
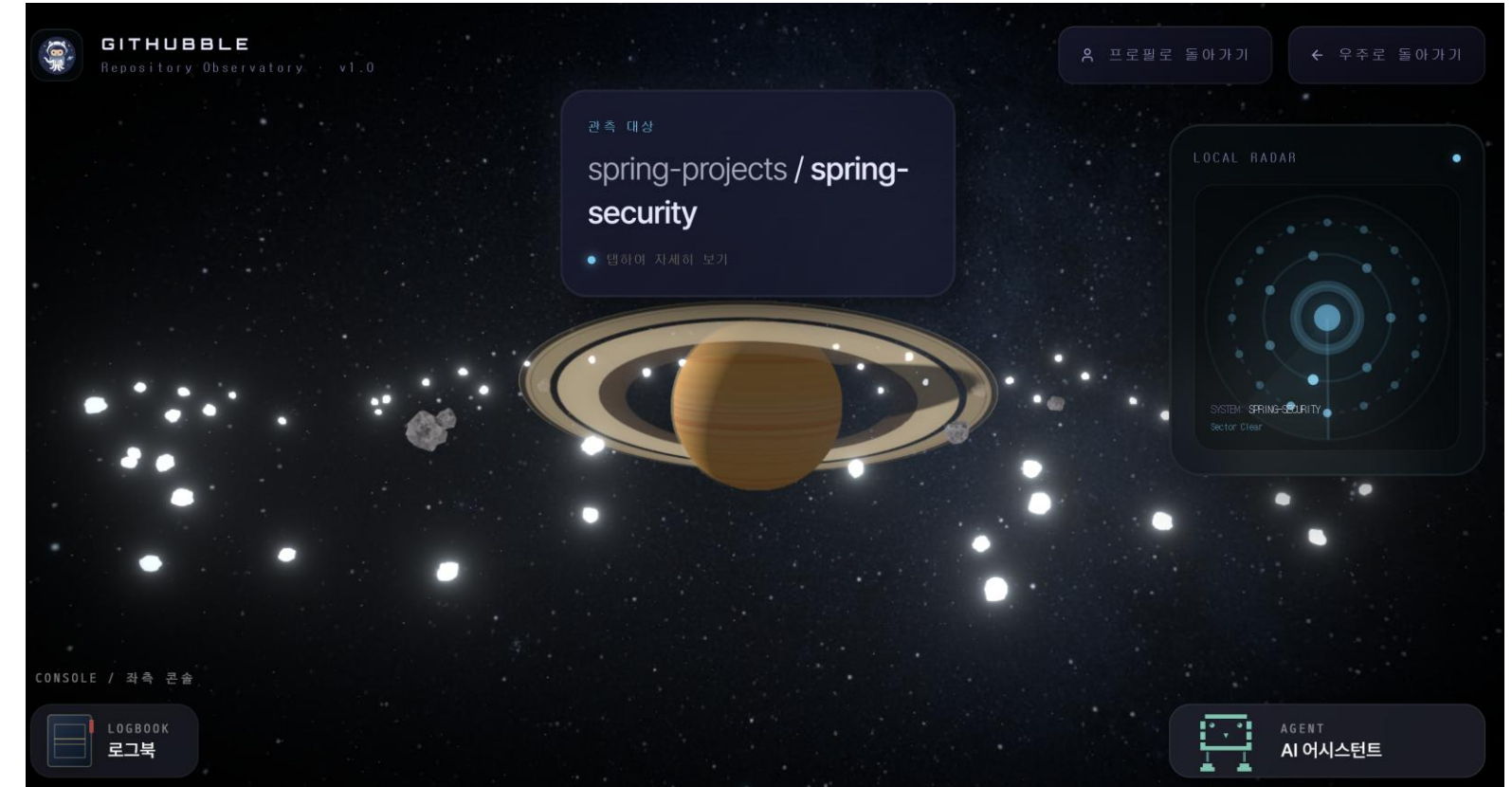
- 서비스: GitHub 저장소를 파싱해 모듈 의존성 그래프와 AI 챗봇을 제공하는 서비스
- 기술 스택: Java 21 · Spring Boot · PostgreSQL · pgvector · WebClient · K3s · Grafana · Loki

담당 역할

- BE 3, AI 1, FE 2로 구성된 팀에서 BE를 맡아 개발했습니다.
- AI 코딩을 위한 문서화 환경 세팅, 로그 인프라 구축을 추가로 담당했습니다.

개인 기여도

- AI 코딩 결과물의 품질 유지를 위한 문서화 환경 세팅 (80%)
- GitHub API Parsing 백엔드 구현 (70%)
- 대형 저장소 모듈 분할 그래프 생성(Python) (100%)
- 운영 관측성 인프라 구축 (Grafana / Loki / Prometheus) (40%)
- parse-result/generation 작업의 race condition 해결 (80%)



GitHub — AI 협업 워크플로

■ 프로젝트 내부 문서를 라우팅하고, 안드레 카파시의 CLAUDE.md를 적용했습니다.

- 생각하기: 코딩 전 GOAL plan 작성, 가정·트레이드오프 명시
- 단순함: 문제를 해결하는 최소한의 코드
- 외과수술: patch 도구로 타겟 변경만, 기존 코드 보존
- 목표 실행: 성공 기준 정의 → 로그 증거 기반 반복 → 기록

■ .local — AI 세션 간 지식 베이스 (196 문서)

be/.local/

- ├── goalworkplan/ 17개 — 구현 전 설계·트레이드오프 기록
- ├── goalworkdone/ 18개 — 완료 후 커밋 해시·검증 결과·보안 검토
- ├── records/ 105개 — 세션별 기록·session handoff 문서
- └── monitorsetting/ 28개 — 모니터링 구축 전 과정 타임라인

➡ AI 세션이 바뀌어도 결정 이유·트레이드오프·검증 결과가 유실되지 않도록

AGENTS.md

개발 사항

- 작업을 시작하기 전에 반드시 be/docs/adr위치의 README.md 와 be/docs/conventions위치의 README.md를 확인하라.
- 각 작업은 가능한 변경된 파일 수가 10개를 넘기지 않도록 일정 단위로 나누어 중간에 커밋하라
- 작업을 마친 후 be/docs/convention/COMMIT-CONVENTION.md를 확인하여 커밋하라

백엔드 문서 디렉토리

be/AGENTS.md

be/CLAUDE.md // AGENTS.md와 같은 내용

be/docs/project.md - 프로젝트 개요 및 구현기능 문서

be/docs/adrs/README.md - ADR목록 요약, 각 문서가 무엇을 다루는지

be/docs/adrs/README.md - CONVENTION목록 요약, 각 문서가 무엇을 다루는지

be/.local - 개인별 문서, 메모, 기록들

1. 코딩하기 전에 생각하기

추측하지 말 것. 혼란을 숨기지 말 것. 트레이드오프를 드러낼 것.

구현하기 전에:

- 가정을 명시적으로 말하세요. 불확실하면 질문하세요.
- 여러 해석이 가능하다면 그것들을 제시하세요. 조용히 하나를 골라 진행하지 마세요.
- 더 단순한 접근이 있다면 말하세요. 필요하다면 이의를 제기하세요.
- 무언가 불명확하다면 멈추세요. 무엇이 헛갈리는지 이름 붙여 말하고 질문하세요.

2. 단순함 우선

문제를 해결하는 최소한의 코드. 추측성 구현은 하지 말 것.

- 요청받은 것 이상의 기능을 추가하지 마세요.
- 한 번만 쓰이는 코드에 추상화를 만들지 마세요.
- 요청되지 않은 “유연성”이나 “설정 가능성”을 추가하지 마세요.
- 발생할 수 없는 시나리오를 위한 예외 처리를 넣지 마세요.
- 200줄로 작성했는데 50줄로도 가능하다면 다시 작성하세요.

GitHubble — 운영 관측성 구축

2026.04 ~ 2026.05 (인원 : 6명)

■ 모니터링 스택 구성

레이어 구성 요소 역할

메트릭 Actuator→Prometheus→Grafana JVM·HTTP 응답시간·에러율

로그 Grafana Alloy→Loki→Grafana 앱 로그 수집·검색

추적 generationRunId BE·GD 로그 연결

보안 AccessLogFilter+LogSanitizer PAT·JWT·API key 마스킹

■ generationRunId — BE·GD 로그 연결

- BE에서 generation 요청 시 generationRunId 생성 → GD Python 전파
- 장애 시 BE·GD 로그를 같은 ID로 동시 추적 가능



```
[*] techstacks=[Python, WebSocket, Docker, GitHub Actions] elapsedms=0
2026-05-28 17:26:53 2026-05-28T08:26:53.963Z INFO 1 --- [be] [pool-5-thread-3] .d.g.s.GitHubRepositoryGenerationService :
generation.event event=generation_stage_started eventName=생성 단계 시작 ownerRepo=Kludex/uvicorn genera
tionRunId=7f9c3a8e-6335-4e94-9ff6-452410e96557 owner=Kludex repo=uvicorn appuserId=6 commitSha=57b3991
d5f6f8e2f750378be5eada7c4fedd90f9 stage=tech_stack_save stageName=기술 스택 저장
2026-05-28 17:26:53 2026-05-28T08:26:53.966Z INFO 1 --- [be] [pool-5-thread-3] .d.g.s.GitHubRepositoryGenerationService :
Resolving tech stack mappings: repoId=109, requestedCount=4, distinctCount=4, resolvedCount=4, request
edNames=[Python, WebSocket, Docker, GitHub Actions], reason=Tracing existing tech stack mapping upsert
inputs without changing mapping behavior
2026-05-28 17:26:53 2026-05-28T08:26:53.969Z INFO 1 --- [be] [pool-5-thread-3] .d.g.s.GitHubRepositoryGenerationService :
Applying tech stack mapping changes: repoId=109, existingCount=2, deleteCount=0, saveCount=2, reason=T
racing existing delete/save counts before repository mapping persistence
```

GitHubble — Race Condition 해결

2026.04 ~ 2026.05 (인원 : 6명)

■ 문제

- FE가 /parse-result와 /generation을 동시에 호출
- 동일 github_repo_id를 두 스레드가 동시에 INSERT
- UNIQUE constraint 위반 → 500 에러 발생

■ 원인 분석

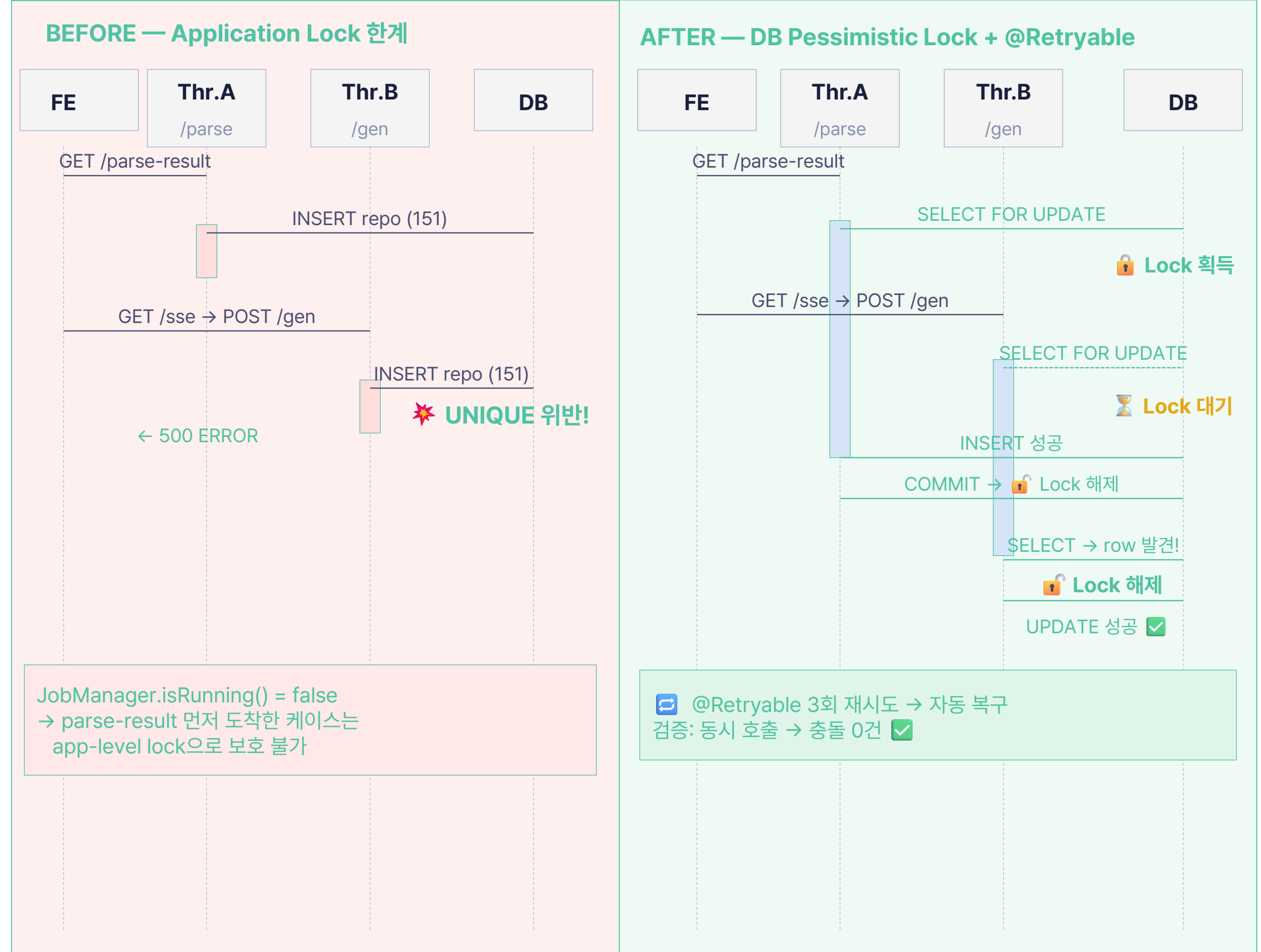
- JobManager의 application-level lock 확인
- 동일 JVM 내에서만 유효
→ 다른 스레드 간 보호 불가

■ 의사 결정

- optimistic lock: 충돌 후 롤백·재시도
- pessimistic lock 선택: optimistic은 insert 충돌 시 사용자 경험이 불편하고, pessimistic이 재시도 비용 작음

■ 해결

- findByGithubRepoldWithLock()
@Lock(PESSIMISTIC_WRITE)
- @Retryable(DataIntegrityViolationException)
재시도
- spring-retry + AOP 의존성 추가



문제 상황 — 단일 LLM 호출의 한계

facebook/react (6,866 files)



왜 실패했나?

토큰 초과

6,866개 파일 트리를
한 번에 입력 → LLM의 컨텍스트 한계 초과

출력 잘림

max_output=6,500
JSON 중간에 끊김

구조적 한계

모델 교체(gpt-5.2 -> DeepSeek-V4 Pro)
로도 해결 불가

(근본 원인은 파일 수)

해결 전략 — 모듈 분할 + 병렬 처리

INPUT

6,866 files · 644K chars

```
.claude/instructions.md
.claude/settings.json
.github/workflows/ci.yml
compiler/.eslintrc.js
compiler/CLAUDE.md
compiler/apps/playground/
  components/Editor.tsx
  components/Output.tsx
compiler/packages/
  babel-plugin-react-
  compiler/src/HIR/...
packages/react-dom/src/
packages/react-reconciler/
packages/react-devtools-
  shared/src/backend/...
scripts/rollup/build.js
scripts/jest/config.js
yarn.lock
... (6,866줄)
```

STEP 1 condense

644K → ~36K chars (↓94%)

```
# depth=3 집계 결과

compiler/packages/
  babel-plugin-react-
  compiler/ (3,654 files)
compiler/apps/
  playground/ (65 files)
packages/react-devtools-
  shared/src/ (609 files)
packages/react-dom/src/
  (185 files)
packages/react-reconciler/
  src/ (166 files)
scripts/jest/ (17 files)
scripts/rollup/ (17 files)
fixtures/dom/src/ (90 f)
...

→ ~792그룹, 36K chars
```

644K → 36K

chars 압축 (94%↓)

STEP 2 detect + split

7개 감지 → 11개 모듈

```
# 7개 감지

compiler      3,836 ⚠ >500
fixtures      566 ⚠ >500
pkg/devtools  614 ⚠ >500
pkg/react-dom 223 ✅
pkg/reconciler 174 ✅
scripts       197 ✅

# 재귀분할 후 11개

__tests__    3,452 ⚠ flat
fixtures_core 465 ✅
devtools_hooks 250 ✅
devtools_ui   244 ✅
react-dom    223 ✅
scripts      197 ✅
reconciler   174 ✅
babel_core   129 ✅

→ 총 11개 모듈
```

6,866 → 11

files → 모듈

STEP 3 asyncio.gather

11개 동시 LLM → 병합

```
# 11개 동시 LLM 호출

asyncio.gather(
  gen(mod[0]), # ~7분 ⚠
  gen(mod[1]), # ~45초
  gen(mod[2]), # ~38초
  ...         # 11개
)

# 병합 결과

34개 노드
48개의 간선
10개의 그룹

→ DiagramGraph
→ Mermaid 컴파일
→ S3 저장
→ Neo4j ingest

✅ 7.5분 완료
```

20분 → 7.5분

4/4 대형 저장소 성공

검증 결과, 회고

저장소	파일 수	분할 전	분할+병렬 후	개선
facebook/react	6,866	20분 + 실패 ❌	~7.5분 ✅	▼ 62%
python/cpython	6,079	36분 + 실패 ❌	~5.0분 ✅	▼ 86%
apache/tomcat	5,348	32분 + 실패 ❌	~7.6분 ✅	▼ 76%
apache/kafka	9,035	48분 + 실패 ❌	~7.7분 ✅	▼ 84%

■ 프로젝트 회고

- AI 모델이 가진 한계를 SW 엔지니어적으로 해결하고자 노력했습니다. 압축 및 분할 요청을 통해 거대한 저장소도 다이어그램으로 나타낼 수 있었습니다.
- 그러나 한편 분할이 되면서 모듈단위의 분석은 가능했지만, 세부 파일에 대한 세세한 분석은 어려운 점이 있었습니다. 어떤 모듈이 있는지는 알 수 있었으나 그 모듈에 어떤 파일이 어떤 역할을 하는지는 파악하기 어렵다는 단점이 있었으며, 추후 이를 해결해 나가고자 합니다.

지역축제 정보제공 서비스 PAMMAP

2026.01 ~ 2026. 02 (인원 : 6명)

프로젝트 개요

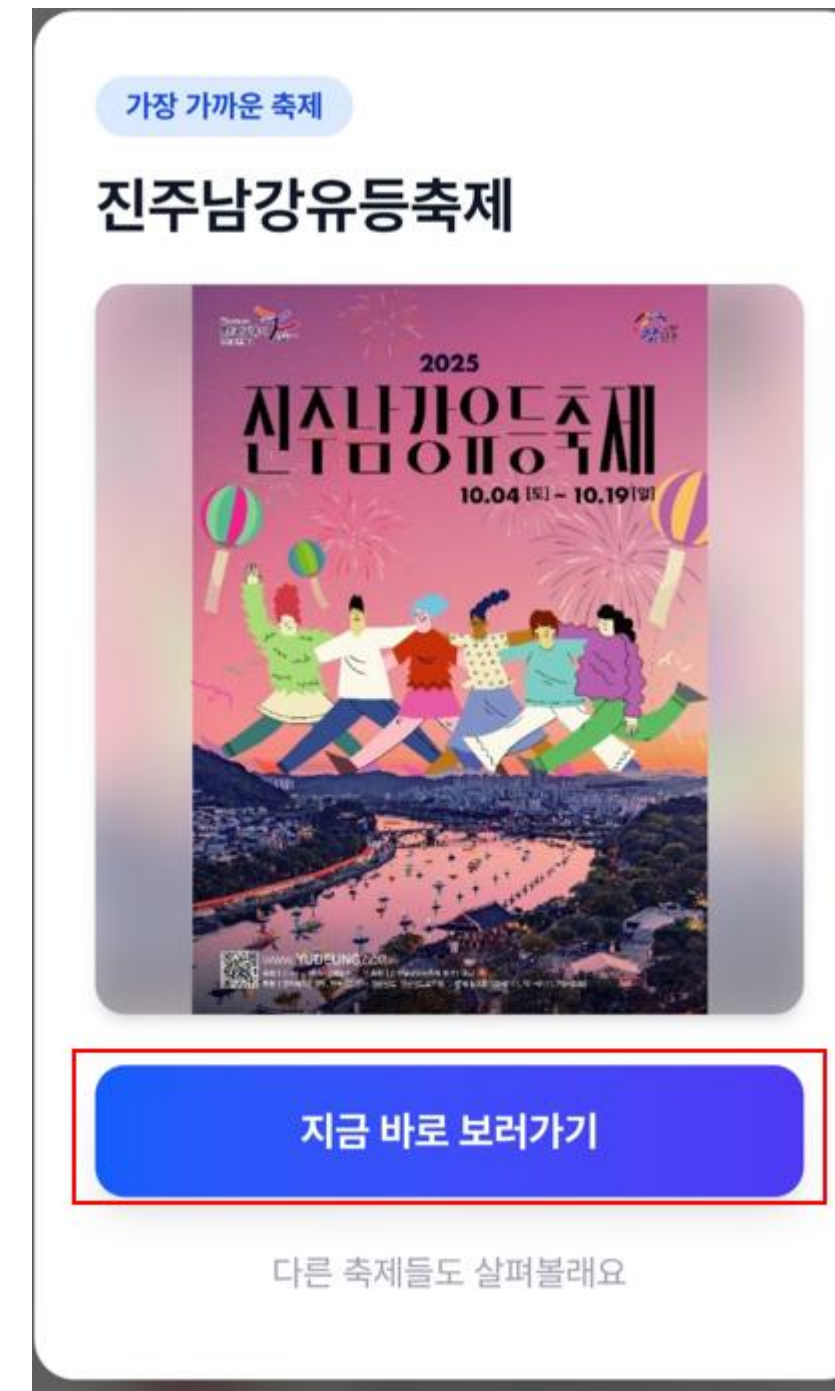
- 목표 : 주요 축제를 선정하여, 손 안의 팸플릿처럼 편하게 정보 보기
- 배경 : 순수 개발시간 2주 이내에, 부족한 축제관련 데이터를 보충하기 위해 수동으로 데이터 세팅 작업을 실시해야 했습니다.
- 주요 기능 : 지도앱으로 편하게 보는 것과 같이 지역축제의 정보를 편하게 보는 것입니다.

담당 역할(역할 작성)

- BE리딩을 맡아 기술문서 작성. 컨벤션 지정. BE개발을 맡았습니다.
- BE 3
- FE 3

개인 기여도

- 축제 필터링 및 검색 기능 구현 (80%)
→ 다양한 조건 기반으로 축제를 탐색할 수 있는 검색 기능 개발
- 사용자 위치 기반 축제 조회 기능 구현 (90%)
→ 현재 위치 기준으로 가까운 축제를 제공하는 기능 개발



처음 서비스에 진입하면, 지리적으로 가까운 축제가 제공됩니다. 지금 바로 보러가기를 누르면 축제 지도 UI가 입혀진 지도 화면으로 이동합니다.

■ 사용 기술 / 적용 맥락 및 이유

- JPA / QueryDSL

→ JDBC, MyBatis와 비교하여 객체 중심 설계 가능, 생산성을 위해 JPA도입. 축제 검색/필터링에서 동적 쿼리를 위한 QueryDSL

- Spring Actuator + Micrometer + Prometheus + Grafana

→ API 서버의 성능 및 상태를 실시간 모니터링 하기 위한 요청 수, 응답 시간, JVM상태 등 핵심 지표를 수집하여 병목구간 파악. Grafana 대시보드로 시각화

- MySQL

→ 지역 축제에 관한 정보가 구조가 명확하여 RDB 선택
팀원 대부분이 익숙했으며, MySQL의 인덱스의 동작방식을 이해하고있어서 성능 개선점을 찾기 편했습니다.

■ 구현 사항

[실시간 인기축제 Top N 조회]

- 현재 진행 중인 축제를 우선하고, 방문객 수 기준으로

상위 N개를 조회하는 API 구현

- QueryDSL을 활용해 진행 여부 + 방문객 수 기반 복합 정렬 로직 설계

[축제 동적 검색 / 필터링]

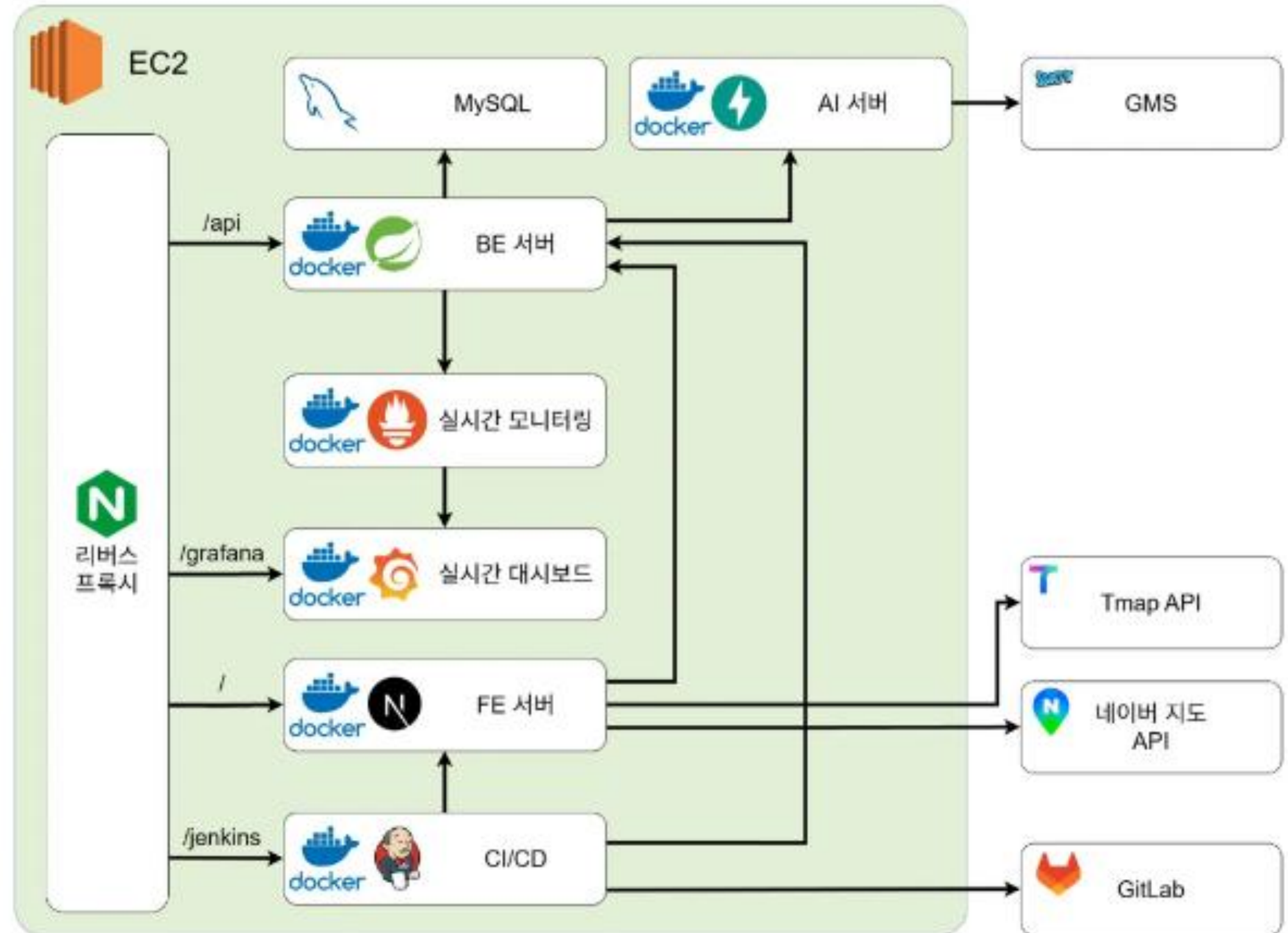
- 지역, 월, 검색어 조건을 조합하여 축제를 유연하게 검색하는 API 구현

- QueryDSL 기반 동적 where절 구성으로 확장 가능한 검색 구조 설계.

[사용자 위치 기반 주변 축제 조회]

- 사용자 위/경도를 기준으로 가까운 축제를 거리순으로 조회하는 API 구현

- Haversine 공식으로 거리 계산 후 QueryDSL로 정렬, 진행 중 필터 옵션 지원

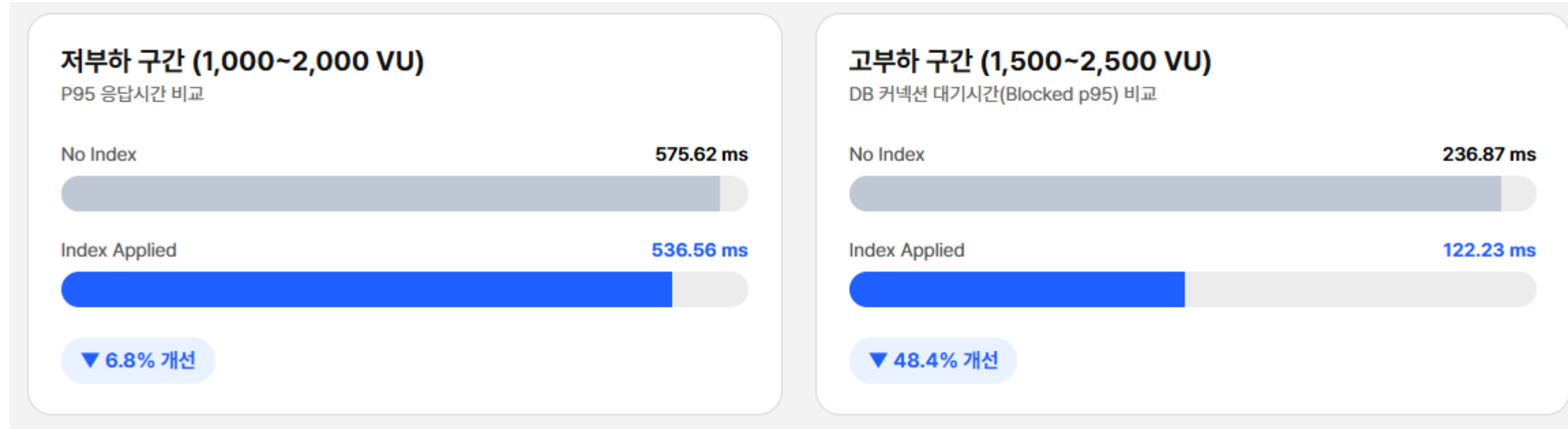


■ 문제 해결 사례

1. 오류 분석/문제 분석

- 문제 현상 : 위치기반 주변축제 조회기능(/nearby) 에 대한 성능 테스트에서 동시 요청 증가 시 응답 지연과 DB 대기가 커졌습니다.
- 원인 : 테스트에 사용된 /nearby 기능이 인덱스를 타지 못했다고 생각했습니다.

2. 개선 과정



3. 해결 과정

- 1) 문제 진단 : 구현 방법을 정할 때 /nearby가 인덱스를 잘 타지 못할거라고 생각하고, 추후 고도화 문서에 남겨뒀습니다. 기능 구현이 마무리 되고 Explain으로 살펴보니 **테이블 풀 스캔(ALL)**이 발생했습니다.
- 2) 전략 수립 : 기존 기능은 **전체 축제를 가져와서 거리 계산 후 정렬하는 구조**였고, 실제 서비스 예시들을 살펴봤을 때, 네이버 지도의 경우 "카페"를 검색하면 반경 수 Km의 결과만 가져옵니다. 마찬가지로 **bounding box로 후보군을 줄이고**, 동일한 거리 계산을 수립했습니다.
- 3) 기술 적용 : (위도, 경도, 축제id) 순으로 인덱스를 설정했습니다.
- 4) 검증 : 임의의 3곳의 위치에서 EXPLAIN ANALYZE를 비교한 결과, 기존 ALL(테이블 풀 스캔)이 range스캔으로 변했고, ROWS_EXAMINED도 79~93% 가량 감소했습니다.

■ PAMMAP 추가 활동 - 문서화 및 회고

[ADR-2] DTO 구현 전략

ADRs	Authors
[ADR-1] API 응답 및 에러 처리 표준화	@영헌 곽
[ADR-2] DTO 구현 전략	@영헌 곽
[ADR-3] 예외 처리 및 로깅 전략	@영헌 곽
[ADR-4] 모니터링 및 성능 측정 전략	@영헌 곽
[ADR-5] Swagger 문서화 전략	@영헌 곽

Context

- Java 16부터 불변형 객체이자 DTO 지원을 위한 Record가 추가됨
- Record vs Class 어떤 구현방식을 사용할 지가 쟁점
- 사람마다 DTO 작성 스타일이 다름 (누구는 `@Data`, 누구는 `@Getter + @Builder`).
- 단순 조회(Read)가 많은 서비스 특성상, 데이터가 중간에 변조되지 않음을 보장하는 불변성(Immutability)이 중요함.

Decision

- Controller와 Service 계층을 오가는 모든 Request/Response DTO는 Java `record` 를 사용하여 구현한다.
- JPA Entity는 기존대로 `class` 를 사용하되, DTO는 무조건 `record` 로 통일한다.
- 데이터 매핑(Entity ↔ Record)은 Record 내부에 `from` (Entity -> DTO) 또는 `toEntity` (DTO -> Entity) 메서드를 두어 처리한다.

상세 규격

- 지양해야 할 방식

```
// 팀원마다 스타일이 갈리고, 코드가 길어짐
@Getter
@NoArgsConstructor
@AllArgsConstructor
@Builder
public class FestivalResponseDto {
    private String name;
    private String location;
    // ... 필드 추가될 때마다 수정 범위가 큼
}
```

- 이렇게 합시다

```
// 선언과 동시에 생성자, Getter, equals, hashCode, toString 자동 생성
public record FestivalResponse(
    String name,
    String location,
    LocalDate startDate,
    LocalDate endDate
) {
    // Entity -> DTO 변환 로직 (Static Factory Method)
    public static FestivalResponse from(Festival entity) {
        return new FestivalResponse(
            entity.getName(),
            entity.getLocation(),
            entity.getStartDate(),
            entity.getEndDate()
        );
    }
}
```

■ 활동 내용 및 회고

- 이번 프로젝트에서는 문서화를 통해 팀원간 정보공유를 원활하게 하고자 했습니다.
- >BE 리딩을 맡아서 주요 구현사항들에 대해 위와 같이 ADR형태로 만들어 공유했습니다.

- 이 프로젝트에서는 문서화를 통한 시개발의 품질 관리와 협업 과정에서의 불필요한 소통을 줄이고자 했습니다.

부동산 청약정보 제공 서비스 Nesto

2025.12 ~ 2025. 12 (인원 : 2명)

프로젝트 개요

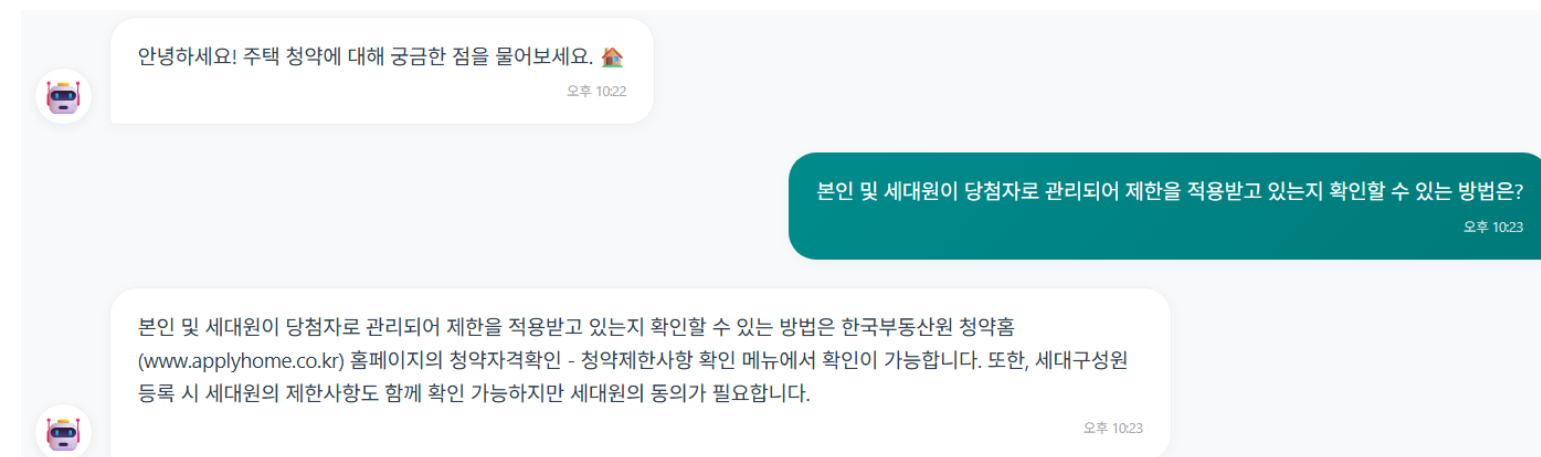
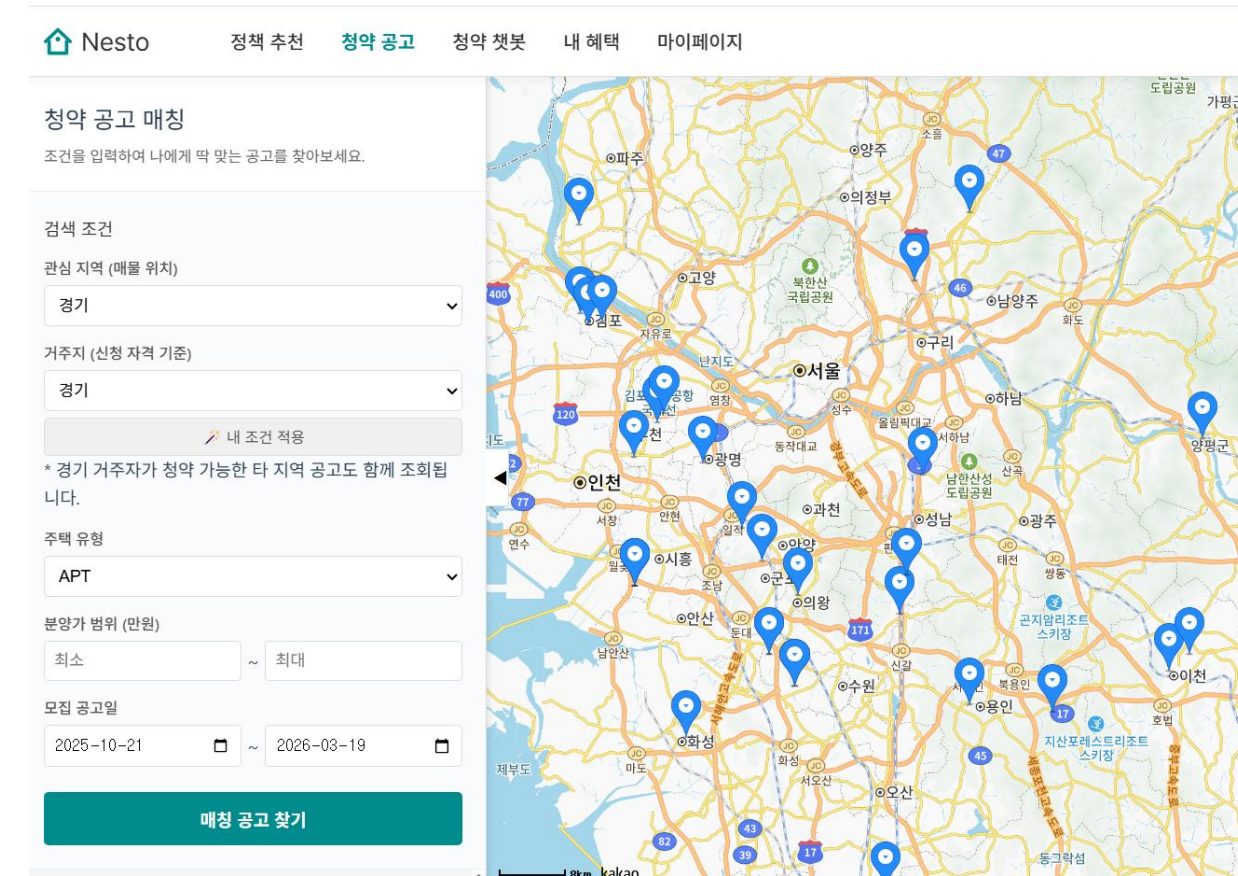
- 목표 : 청년층을 위한 부동산 청약공고 정보 제공
- 배경 : 프로젝트 중반부에 팀원이 이탈하여, 혼자 진행했습니다.
- 주요 기능 : 과거~현재의 청약 매물들을 지도에서 확인 가능, 청약공고 QnA를 학습한 RAG챗봇과의 대화 가능

담당 역할(역할 작성)

- Java Spring, Vue.js로 풀스택 개발을 진행 했습니다.
- 기획 이후 개발단계에서 팀원이 이탈하여 혼자 개발했습니다.

개인 기여도

- 청약공고 API 통합 스키마 구성 및 지도에 표시하기(100%)
 - 다양한 조건으로 필터링 후 지도에 표시 하는 기능 개발
- 청약공고 QnA 정부자료를 학습한 RAG챗봇 개발(100%)
 - 확실한 출처의 자료를 읽어와서 정확한 대답을 하는 챗봇 개발



Q 123 본인 및 세대원이 당첨자로 관리되어 제한을 적용받고 있는지 확인할 수 있는 방법은?

A 한국부동산원 청약홈(www.applyhome.co.kr) 홈페이지의 청약자격확인 - 청약제한사항 확인 메뉴에서 확인이 가능합니다.

* 세대구성원 등록 시 세대원의 제한사항도 함께 확인 가능(세대원의 동의 필요)

■ 사용 기술 / 적용 맥락 및 이유

- Java Spring AI / ChromaDB

- 비교적 최근 출시한 Spring AI의 시험 적용에 가까웠습니다. BE, AI를 같이 개발할 수 있다면 관리가 편해질 것이라고 생각했고, 실제로도 그러했습니다.
- 데이터 규모가 크지 않았고, FAISS, Qdrant를 쓰기보다는 쓰기 편하고, 빠르게 적용하는 것이 중요하다고 생각하여 ChromaDB를 선택했습니다.

- MyBatis

- 청약 매물을 검색할 때 지역, 날짜, 주택 유형, 분양가 등 여러 요소로 동적 쿼리를 작성해야 했는데, JPA보다 편리했습니다.

- Vue.js

- 백엔드 개발을 주로 해서, 진입장벽이 낮은 프레임워크를 찾다가 사용하게 되었습니다. 구조가 명확해서 컴포넌트 기반의 개발을 직관적으로 이해할 수 있었습니다.

■ 구현 사항

[복잡한 필터링 로직으로 청약정보 지도에 띄우기]

- 지역, 주택 유형, 접수 일정, 소득 기준 등 다양한 조건을 조합해 청약 공고를 조회하고, 결과를 지도 UI와 연동했습니다.
- MyBatis Dynamic SQL의 <if>, <choose>, <foreach>를 활용해 동적 쿼리를 구성했고, 관심 등록 여부 LEFT JOIN과 가격 범위 MIN/MAX 조회를 단일 쿼리에서 함께 처리했습니다.

[청약공고 PDF의 RAG챗봇 구현]

- 청약 공고 PDF를 벡터화해 질문에 근거 기반 답변을 제공하는 RAG 챗봇을 구현했습니다.
- SpringAI와 ChromaDB로 임베딩, 검색 파이프라인을 구성했고, QnADocumentSplitter를 구현하여 검색 정확도를 높였습니다.

■ 문제 해결 사례

1. 오류 분석/문제 분석

- 문제 현상 : 공공 API 10종이 주택 유형별로 분리되고, 컬럼명도 불일치했습니다.
- 해결 : 공통/부분공통/단독 컬럼 기준으로 통합 스키마 2개로 설계 했습니다.
- 이유 : 10종을 그대로 테이블로 변환한다면, 불필요한 Join연산이 발생할 수 있다고 판단했습니다. 주택 유형별 스키마를 보수적으로 합집합 개념으로 접근하여 통합했습니다.

■ 다. 상세기능 목록

번호	API 명(국문)	상세기능명(영문)	상세기능명(국문)
1	청약홈 분양정보 조회 서비스	getAPLttotPblancDetail	APT 분양정보 상세조회
2	청약홈 분양정보 조회 서비스	getUrbtyOfctlTtotPblancDetail	오피스텔/도시형/민간임대/생활속박시설 분양정보 상세조회
3	청약홈 분양정보 조회 서비스	getRemndrLttotPblancDetail	APT 잔여세대 분양정보 상세조회
4	청약홈 분양정보 조회 서비스	getAPLttotPblancMdl	APT 분양정보 주택형별 상세조회
5	청약홈 분양정보 조회 서비스	getUrbtyOfctlTtotPblancMdl	오피스텔/도시형/민간임대/생활속박시설 분양정보 주택형별 상세조회
6	청약홈 분양정보 조회 서비스	getRemndrLttotPblancMdl	APT 잔여세대 주택형별 상세조회
7	청약홈 분양정보 조회 서비스	getPblPvtRentLttotPblancDetail	공공지원 민간임대 분양정보 상세조회
8	청약홈 분양정보 조회 서비스	getPblPvtRentLttotPblancMdl	공공지원 민간임대 분양정보 주택형별 상세조회
9	청약홈 분양정보 조회 서비스	getOPTLttotPblancDetail	임의공급 분양정보 상세조회
10	청약홈 분양정보 조회 서비스	getOPTLttotPblancMdl	임의공급 분양정보 주택형별 상세조회

노란색 칼럼
> 5개 공통으로 나타난 칼럼

파랑색 칼럼
> 2개 이상 나타난 칼럼

초록색 칼럼
> 한 번만 나타난 칼럼

그리하여, 오른쪽과 같은 통합 테이블이 나타났고, 나머지 5개에 대해서도 유사한 방식으로 진행하여 총 2개의 청약 표준 테이블을 도출했습니다.

문제 해결 사례

2. 오류 분석/문제 분석

- 문제 현상 : 위도/경도에 나타난 Null값이 41%에 달해 지도상에 제대로 표기할 수 없는 문제

- 원인 : 카카오맵 API를 통해 주소를 넣고 위도/경도 값을 받아오는데, "행정중심복합도시 5-1 생활권 L1블록(세종특별자치시 합강동 일원)" 과 같은

경우 NULL이 되게 됩니다.

- 해결 : 행정중심복합도시 5-1 생활권 L1블록(세종특별자치시 합강동 일원) 과 같은 주소가 있는 경우, NULL이 나오면 괄호 안쪽을 파싱하여 넣어보고, 그래도 NULL이 나온다면 주소 문자열을 N개의 블록으로 나누고, 모든 N블록, N-1개의 블록.... 순으로 넣어보며 NULL에 대한 Fallback 로직을 구현했습니다. 그 결과 NULL 값 비율을 41% -> 0.6%로 감소시켰습니다.

Y	Z	AA	BC	BD	BE	
1	gnrl_rcept_rcptde	gnrl_rcept_bgnde	hssply_adres	plt_plc_geo_la	plt_plc_geo_lo	
2		인천광역시 중구 운남동 1696-1(영종하늘도시 A20BL)	37.47537042	126.5205413		
3		인천광역시 중구 운남동 1695-2(영종하늘도시 A19BL)	37.47703448	126.5195644		
4		경기도 용인시 처인구 역북동 811번지 (신대지구 A1 블록)	37.23979466	127.1880388		
5		부산광역시 동래구 안락동 1230번지 일원	35.19991117	129.1095539		
6		행정중심복합도시 5-1생활권 L1블록(세종특별자치시 합강동 일원)			0.4128195	
7		경기도 용인시 수지구 풍덕천동 71-1번지 일원	37.32968741	127.1023579		
8		경상북도 안동시 송현동 529-1번지 일원	36.5683689	128.6924025		
9		대전광역시 서구 탄방동 514-360번지 일원	36.34242386	127.3805987		
10		울산광역시 중구 반구동 554-5번지 일원	35.55228264	129.3470303		
11		울산광역시 남구 아음동 829-20 일원	35.52329938	129.3299208		
12		울산광역시 남구 아음동 830-1 일원	35.5242218	129.3303777		
13		인천광역시 중구 중산동 1958-8.9 (인천광역시 중구 중산동 RC4-1,2BL)				
14		대전광역시 서구 탄방동 514-360번지 일원	36.34242386	127.3805987		
15		경기도 이천시 중리지구 B3블록(중리동 518번지)				
16		부산광역시 남구 대연동 455-25번지	35.13665066	129.1059736		
17		경기도 용인시 처인구 양지면 양지리 산97-12번지 일원	37.23249114	127.2901952		
18		서울특별시 강남구 역삼동 758번지 일원	37.49745045	127.0480144		
19		인천광역시 연수구 송도동 543번지(송도국제도시 11공구 RC3BL)	37.37510507	126.6829723		
20		경기도 군포시 대야미동, 숙달동, 둔대동 일원 (수원당수 공공주택지구 내 A-2블록				
21		경상남도 창원시 성산구 신월동 90번지 일원	35.2225955	128.6916172		
22		경기도 시흥시 시흥거모 공공주택지구 B-6BL				
23		충청남도 홍성군 홍북읍 신경리 929번지 (충청남도 내포신도시 RH-14BL)	36.65271286	126.6756528		
24		충청남도 홍성군 홍북읍 신경리 929번지 (충청남도 내포신도시 RH-14BL)	36.65271286	126.6756528		
25		경기도 수원시 권선구 당수동 3001번지 일원 (수원당수 공공주택지구 M1BL)				
26		경기도 수원시 권선구 당수동 3002번지 일원 (수원당수 공공주택지구 M2BL)				
27		경기도 이천시 증포동 107-2번지 일원	37.28845454	127.4612908		
28		경기도 남양주시 진접읍, 진건읍, 퇴계원을 일원 남양주왕숙 공공주택지구 내 A-24블록				
29		경기도 남양주시 진접읍, 진건읍, 퇴계원을 일원 남양주왕숙 공공주택지구 내 B-17블록				
30		제주특별자치도 제주시 연동 282-4 외 4필지				
31		인천 검단산도시 AB13블록 (인천광역시 서구 원당동 1063-2 일원)				
32		제주특별자치도 제주시 도련일동 2878				
33		경기도 부천시 오정구 여월동 8-9번지 외 4필지				
34		충청북도 청주시 상당구 지북동 일원 청주지북 공공지원민간임대주택 공급촉진지구 내 B1블록				
35		전라남도 나주시 이창동 714 일대	34.99507747	126.7033603		
36		울산광역시 남구 아음동 388-7번지 일원	35.5219348	129.3341372		
37		울산광역시 남구 아음동 389-49번지 일원	35.52094082	129.3335842		
38		충청남도 천안시 청당동 310-6번지 일원(1BL-1Lot)	36.77661567	127.1479103		
39		경기도 김포시 북변동 224-67 일원	37.62508724	126.705401		
40		경기도 화성시 남양읍 남양리 2198 (남양뉴타운 B16BL)	37.20748902	126.838168		
41		세종특별자치시 합강동 5101-9번지 일원(행정중심복합도시 5-1생활권 L9BL)				
42		인천광역시 미추홀구 학익동 인천 용현·학익 1블록 도시개발구역 공동2BL				
43		경기도 의왕시 고천동 265번지 일원	37.34759741	126.9760334		
44		경기도 안양시 만안구 안양동 398-32번지 일원	37.38102867	126.9310869		
45		전북특별자치도 순창군 순창읍 순화리 951 번지 일원	35.38155777	127.1374563		
46		충청남도 천안시 동남구 풍서로 801 (용곡동 617번지)	36.78275331	127.143068		
47		경기도 부천시 오정구 원종동 298-1번지 일원	37.52766726	126.80922		
48		경기도 시흥시 시흥거모 공공주택지구 B-2블록(경기도 시흥시 거모동 일원)				
49		경기도 시흥시 시흥거모 공공주택지구 S-2블록(경기도 시흥시 거모동 일원)				
50		울산광역시 울주군 범서읍 서사리 515-2 일원 (울산 다우2지구 B-1BL)	35.59408619	129.2708509		



AA	BC	BD	BE	BF
hssply_adres	plt_plc_geo_la	plt_plc_geo_lo		
인천광역시 중구 운남동 1696-1(영종하늘도시 A20BL)	37.47537042	126.5205413		
인천광역시 중구 운남동 1695-2(영종하늘도시 A19BL)	37.47703448	126.5195644		
경기도 용인시 처인구 역북동 811번지 (신대지구 A1 블록)	37.23979466	127.1880388		
부산광역시 동래구 안락동 1230번지 일원	35.19991117	129.1095539		
행정중심복합도시 5-1생활권 L1블록(세종특별자치시 합강동 일원)	36.52512762	127.3303042		
경기도 용인시 수지구 풍덕천동 71-1번지 일원	37.32968741	127.1023579		
경상북도 안동시 송현동 529-1번지 일원	36.5683689	128.6924025		0.00619914762
대전광역시 서구 탄방동 514-360번지 일원	36.34242386	127.3805987		
울산광역시 중구 반구동 554-5번지 일원	35.55228264	129.3470303		
울산광역시 남구 아음동 829-20 일원	35.52329938	129.3299208		
울산광역시 남구 아음동 830-1 일원	35.5242218	129.3303777		
인천광역시 중구 중산동 1958-8.9 (인천광역시 중구 중산동 RC4-1,2BL)	37.49183628	126.5710648		
대전광역시 서구 탄방동 514-360번지 일원	36.34242386	127.3805987		
경기도 이천시 중리지구 B3블록(중리동 518번지)	37.27226759	127.4350893		
부산광역시 남구 대연동 455-25번지	35.13665066	129.1059736		
경기도 용인시 처인구 양지면 양지리 산97-12번지 일원	37.23249114	127.2901952		
서울특별시 강남구 역삼동 758번지 일원	37.49745045	127.0480144		
인천광역시 연수구 송도동 543번지(송도국제도시 11공구 RC3BL)	37.37510507	126.6829723		
경기도 군포시 대야미동, 숙달동, 둔대동 일원 (수원당수 공공주택지구 내 A-2블록	37.32912671	126.9184082		
경상남도 창원시 성산구 신월동 90번지 일원	35.2225955	128.6916172		
경기도 시흥시 시흥거모 공공주택지구 B-6BL	37.35670182	126.7741749		
충청남도 홍성군 홍북읍 신경리 929번지 (충청남도 내포신도시 RH-14BL)	36.65271286	126.6756528		
충청남도 홍성군 홍북읍 신경리 929번지 (충청남도 내포신도시 RH-14BL)	36.65271286	126.6756528		
경기도 수원시 권선구 당수동 3001번지 일원 (수원당수 공공주택지구 M1BL)	37.2887193	126.940217		
경기도 수원시 권선구 당수동 3002번지 일원 (수원당수 공공주택지구 M2BL)	37.2887193	126.940217		
경기도 이천시 증포동 107-2번지 일원	37.28845454	127.4612908		
경기도 남양주시 진접읍, 진건읍, 퇴계원을 일원 남양주왕숙 공공주택지구 내 A-24블록	37.72630049	127.1898865		
경기도 남양주시 진접읍, 진건읍, 퇴계원을 일원 남양주왕숙 공공주택지구 내 B-17블록	37.72630049	127.1898865		
제주특별자치도 제주시 연동 282-4 외 4필지	33.48941262	126.4923648		
인천 검단산도시 AB13블록 (인천광역시 서구 원당동 1063-2 일원)	37.59394103	126.7215029		
제주특별자치도 제주시 도련일동 2878	33.50560809	126.5892888		
경기도 부천시 오정구 여월동 8-9번지 외 4필지	37.5183234	126.8015516		
충청북도 청주시 상당구 지북동 일원 청주지북 공공지원민간임대주택 공급촉진지구 내 B1블록	36.59496555	127.5087195		
전라남도 나주시 이창동 714 일대	34.99507747	126.7033603		
울산광역시 남구 아음동 388-7번지 일원	35.52094082	129.3335842		
울산광역시 남구 아음동 389-49번지 일원	35.5219348	129.3341372		
울산광역시 남구 아음동 389-49번지 일원	35.52094082	129.3335842		
충청남도 천안시 청당동 310-6번지 일원(1BL-1Lot)	36.77661567	127.1479103		
경기도 김포시 북변동 224-67 일원	37.62508724	126.705401		
경기도 화성시 남양읍 남양리 2198 (남양뉴타운 B16BL)	37.20748902	126.838168		
세종특별자치시 합강동 5101-9번지 일원(행정중심복합도시 5-1생활권 L9BL)	36.52512762	127.3303042		
인천광역시 미추홀구 학익동 인천 용현·학익 1블록 도시개발구역 공동2BL	37.44101799	126.6543808		
경기도 의왕시 고천동 265번지 일원	37.34759741	126.9760334		
경기도 안양시 만안구 안양동 398-32번지 일원	37.38102867	126.9310869		
전북특별자치도 순창군 순창읍 순화리 951 번지 일원	35.38155777	127.1374563		
충청남도 천안시 동남구 풍서로 801 (용곡동 617번지)	36.78275331	127.143068		
경기도 부천시 오정구 원종동 298-1번지 일원	37.52766726	126.80922		
경기도 시흥시 시흥거모 공공주택지구 B-2블록(경기도 시흥시 거모동 일원)	37.35670182	126.7741749		
경기도 시흥시 시흥거모 공공주택지구 S-2블록(경기도 시흥시 거모동 일원)	37.35670182	126.7741749		
울산광역시 울주군 범서읍 서사리 515-2 일원 (울산 다우2지구 B-1BL)	35.59408619	129.2708509		

■ 문제 해결 사례

3. 오류 분석/문제 분석

- 문제 현상 : 청약공고 PDF기반 RAG챗봇에서 질문 의도와 맞지 않는 답변이 발생
- 원인 : 기본 길이 기반 splitter를 사용 시 질문과 답변이 서로 다른 청크로 분리되어 문맥이 끊김
- 한계 : 페이지 단위 또는 고정 길이 분할만으로는 QnA구조와 섹션 문맥을 보존하기 어려웠음

개선 과정

- QnADocumentSplitter를 직접 구현하여 PDF를 페이지별 Document로 읽고 줄 단위로 순회
- 질문 시작 패턴과 섹션 헤더를 정규식으로 탐지하여 질문~답변 묶음을 하나의 청크로 유지
- 청크 생성 시 기존 페이지 메타데이터에 section, category=Q&A를 추가하여 검색 시 문맥 정보를 함께 활용하도록 설계
- 긴 답변은 4000자 기준으로 보조 분할하여 과도한 청크 생성을 방지

```
// 질문 시작 패턴:  
// "Q1. 질문?" 또는 "1 질문?"처럼 번호와 물음표로 끝나는 질문 라인을 감지한다.  
// 질문의 시작점을 기준으로 Q&A 단위 청크를 구성하기 위해 사용한다.  
private static final Pattern QUESTION_PATTERN = Pattern.compile(  
    regex: "^\\s*(?:Q\\s*)?(\\d+)[.\\s]+(.*\\?)\\s*$",  
    Pattern.MULTILINE  
);
```

```
// 섹션 헤더 패턴:  
// "I. 제목", "1. 제목"처럼 문서의 대분류/소분류 제목을 감지한다.  
// 청크 분리 기준으로 바로 사용하지 않고, 현재 문맥(section) 추적용으로 사용한다.  
private static final Pattern SECTION_HEADER_PATTERN = Pattern.compile(  
    regex: "^\\s*([IVX]+|\\d+)\\.\\s+(.*)$",  
    Pattern.MULTILINE  
);
```

```
@Override  
public List<Document> apply(List<Document> documents) {  
    List<Document> splitDocuments = new ArrayList<>();  
  
    String currentSection = "General"; // 현재 섹션 문맥  
    StringBuilder qnaBuffer = new StringBuilder(); // 질문 + 답변을 누적할 버퍼  
    Map<String, Object> currentMetadata = null;  
  
    // 페이지 전체를 먼저 합칠 수도 있지만,  
    // 그 경우 질문이 시작된 페이지 번호 같은 메타데이터가 흐려질 수 있다.  
    // 따라서 페이지를 순회하면서 섹션 상태를 유지하고,  
    // 질문이 시작되는 시점을 기준으로 Q&A 버퍼를 관리한다.  
    for (Document doc : documents) {  
        String content = doc.getContent();  
        String[] lines = content.split(regex: "\\r?\\n");  
  
        for (String line : lines) {  
            // 1. 섹션 헤더 감지: 현재 문맥만 갱신한다.  
            Matcher sectionMatcher = SECTION_HEADER_PATTERN.matcher(line);  
            if (sectionMatcher.find()) {  
                currentSection = sectionMatcher.group(group: 0).trim();  
                logger.debug("섹션 헤더 감지: {}", currentSection);  
  
                // 섹션 헤더는 즉시 청크를 끊는 기준이 아니라,  
                // 이후 생성될 Q&A 청크에 붙일 문맥 정보로만 활용한다.  
            }  
  
            // 2. 질문 시작 라인 감지  
            Matcher qMatcher = QUESTION_PATTERN.matcher(line);  
            if (qMatcher.find()) {  
                // 새 질문이 시작되면, 이전까지 누적한 Q&A를 하나의 청크로 저장한다.  
                if (qnaBuffer.length() > 0) {  
                    splitDocuments.add(createDocument(qnaBuffer.toString(), currentSection, currentMetadata));  
                    qnaBuffer.setLength(newLength: 0);  
                }  
  
                // 새 질문으로 버퍼를 시작한다.  
                qnaBuffer.append(line).append(str: "\\n");  
            }  
        }  
    }  
}
```

분할 예시

원문

I. 청약통장

1. 청약통장이 꼭 있어야 하나요?

청약 유형에 따라 다릅니다.

일반공급에서는 필요한 경우가 많습니다.

- 기본 splitter

[1. 청약통장이 꼭 있어야 하나요?]

[청약 유형에 따라 다릅니다. 일반공급에서는...]

- QnADocumentSplitter

[section=I. 청약통장, category=Q&A]

1. 청약통장이 꼭 있어야 하나요?

청약 유형에 따라 다릅니다.

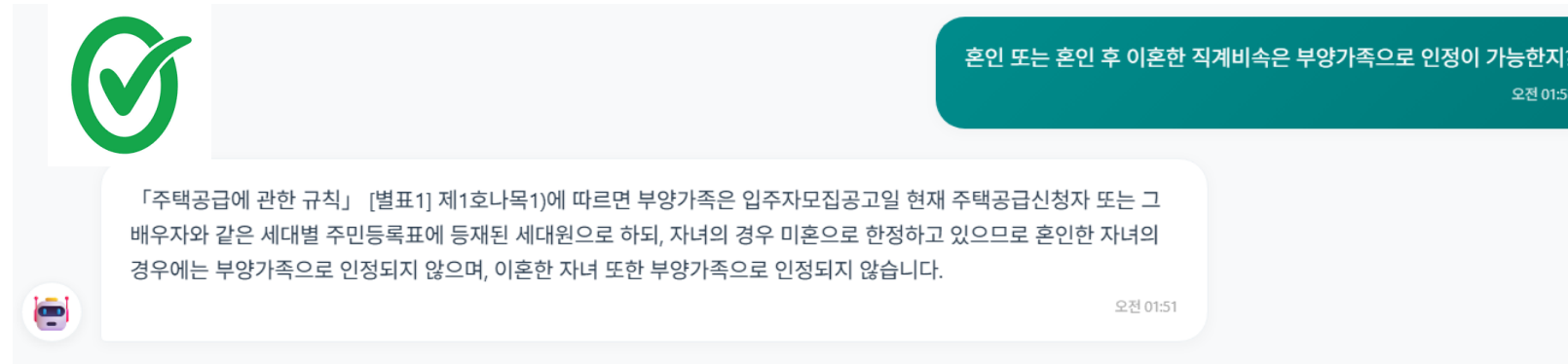
일반공급에서는 필요한 경우가 많습니다.



Q 159 혼인 또는 혼인 후 이혼한 직계비속은 부양가족으로 인정이 가능한지?

A 「주택공급에 관한 규칙」 [별표1] 제1호나목1)에 따르면 부양가족은 입주자모집공고일 현재 주택공급신청자 또는 그 배우자와 같은 세대별 주민등록표에 등재된 세대원으로 하되, 자녀의 경우 미혼으로 한정하고 있으므로 혼인한 자녀의 경우에는 부양가족으로 인정되지 않으며, 이혼한 자녀 또한 부양가족으로 인정되지 않습니다.

FAQ의 질의응답에 해당하는 내용을 찾아 정확히 대답하는 챗봇



■ 프로젝트 성과 및 결과

- SpringAI를 사용해서 프로젝트를 처음으로 진행했는데, 최근에 나와서 활발하게 발전하는 프로젝트라서 AI가 작업할 경우 이전 스냅샷 버전을 사용한다던지. 의존성 문제가 많았습니다.
- RAG를 활용하여 내부 문서를 활용하고자 하는 기업들이 많은데, 이번 시도를 통해 RAG가 단순히 적당히 토큰으로 문서를 쪼개고 VectorStore에 집어넣는게 아니라 굉장히 다양한 변수가 가능하다는 것을 알게 되었습니다.

■ 프로젝트 회고

- 중간에 팀원이 이탈하여 혼자 진행하게 되었고, FE연동, BE작업, AI 작업 모두 맡아 작업했습니다.
- 만약 여기서 더 발전시킨다면, RAG챗봇의 정확도를 더 다양한 방법론을 사용하여 향상시킬 것 같습니다.(TOK-K의 매직넘버 찾기, 또 다른 Split방법 찾기 등)

Java OSS Armeria 기여

2025.08 ~ 2026. 03 (인원 : 1명)

기여 개요

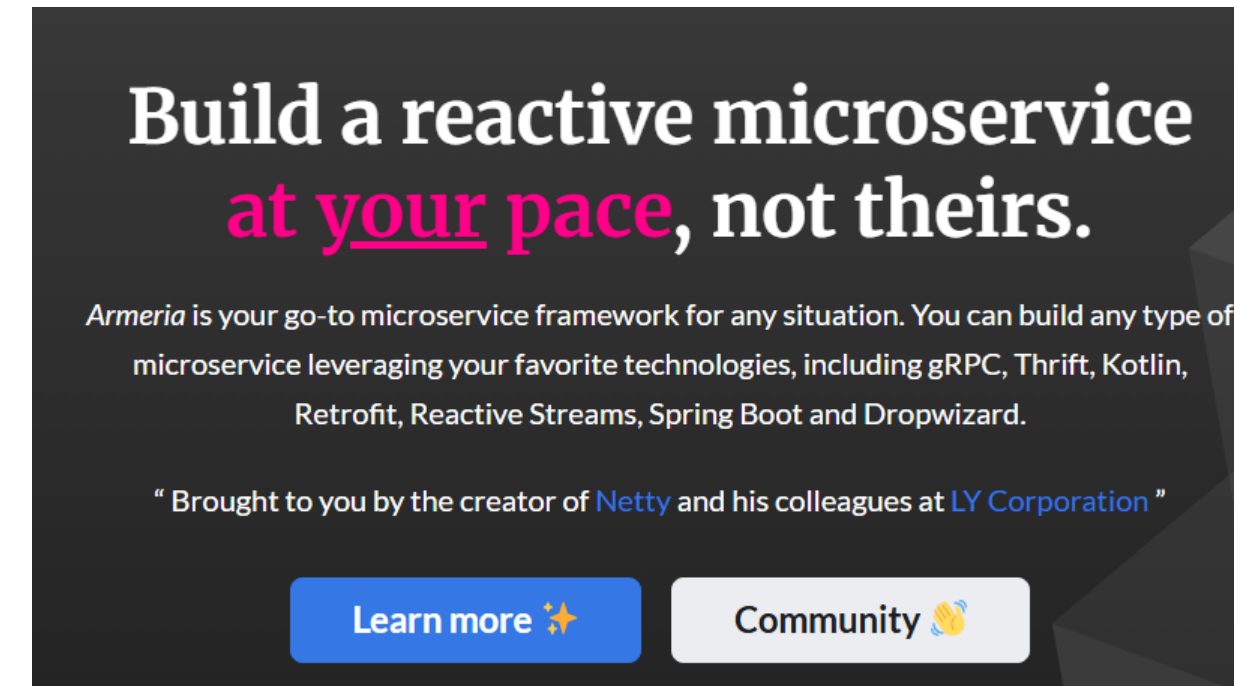
- 대상 : LINE/Armeria 오픈소스 (Start 5000++)
- 주제 : 자체 API문서화 기능인 DocService의 inheritance / polymorphism 문서화 지원
- 기여: Jackson annotation 기반 polymorphism 지원 구현
- 현재 상태 : Merge되어 Contributor가 되었음.

담당 역할

- 이슈 분석 및 구현 방향 제안
- DocService 내부 구조 추적 및 설계 판단
- Provider 구현 및 SPI 등록
- 공통 타입 변환 로직 리팩토링
- PR/Discussion 작성, maintainer 피드백 반영
- gradle parallel build 실패 재현 및 로그 정리 후 보고

개인 기여도

- Jackson기반 다형성, 상속 기능 및 테스트 추가, Schema 구조 개편
- 리뷰 반영 및 빌드 문제 조사
- 문서 렌더링은 Reviewer의 도움을 받음



**Build a reactive microservice
at your pace, not theirs.**

Armeria is your go-to microservice framework for any situation. You can build any type of microservice leveraging your favorite technologies, including gRPC, Thrift, Kotlin, Retrofit, Reactive Streams, Spring Boot and Dropwizard.

" Brought to you by the creator of [Netty](#) and his colleagues at [LY Corporation](#) "

[Learn more](#) ✨ [Community](#) 🍯

feat(docservice): Support Jackson polymorphism annotations #6370

Merged ikhoon merged 13 commits into line:main from YoungHoney:YoungHoney6313 13 hours ago

Conversation 82 Commits 13 Checks 17 Files changed 26 +1,635 -726



YoungHoney commented









Contributor ...

on Aug 26, 2025 • edited

Motivation

This pull request implements support for Jackson's polymorphism annotations (`@JsonTypeInfo`, `@JsonSubTypes`)

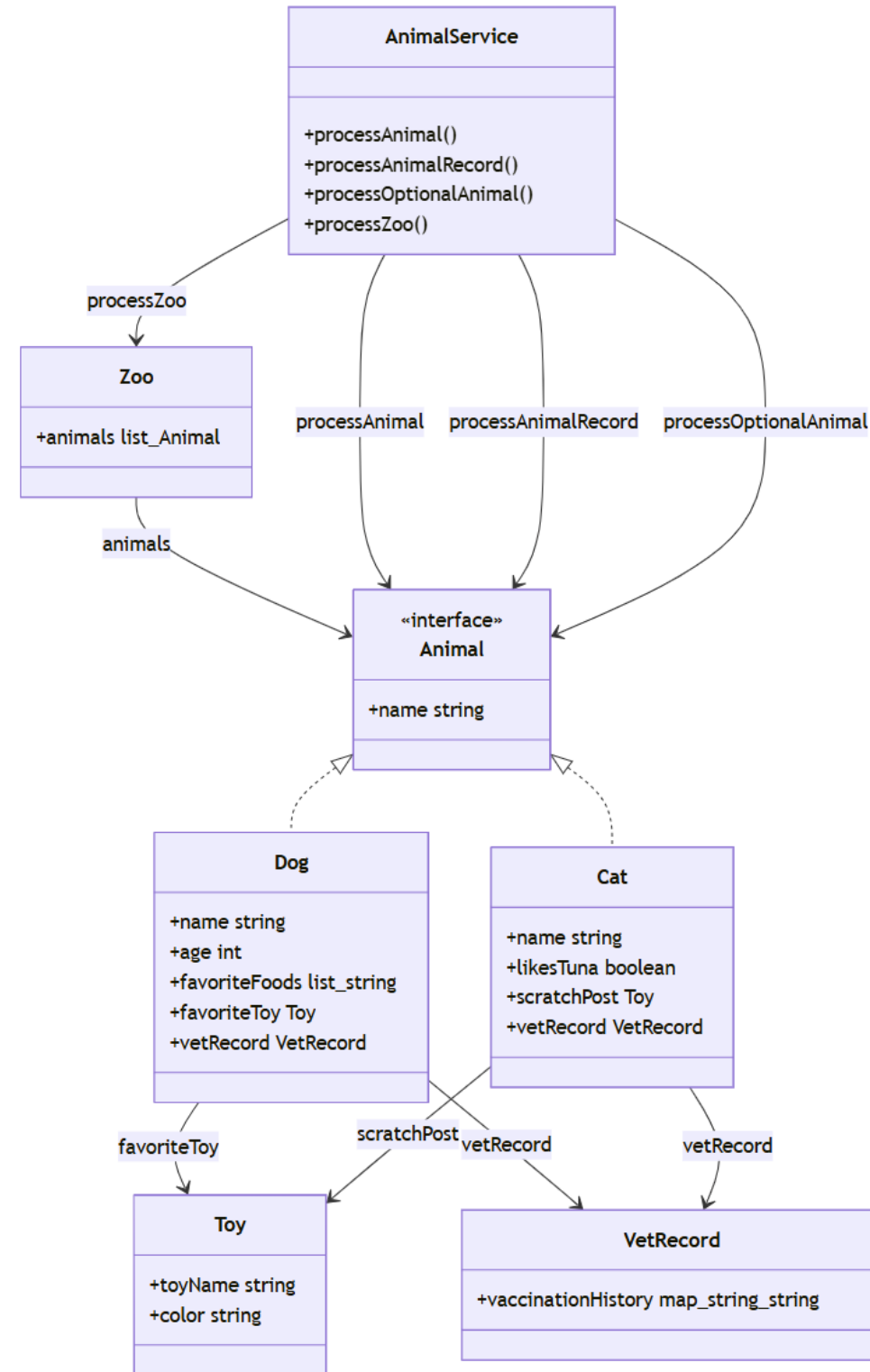
Reviewers

-  coderabbitai[bot] 
-  ikhoon 
-  jrhee17 
-  minwoox 

문제 상황

- Armeria는 Swagger UI와 비슷한 자체 API문서화 기능을 제공하고 있음.

- 그러나, 상속, 다형성 처리가 미흡하여 Armeria 내부 문서로 볼 경우 Structs에 상속 구조의 Dog와 Cat 등이 나타나지 않는 문제가 발생함.



Armeria documentation service 1.37.1

Services

Overview

PolymorphismDocServiceDemo\$AnimalService ^

- POST** processAnimal()
exact:/api/animal
- POST** processAnimalRecord()
exact:/api/animal/record
- POST** processOptionalAnimal()
exact:/api/animal/optional
- POST** processZoo()
exact:/api/zoo

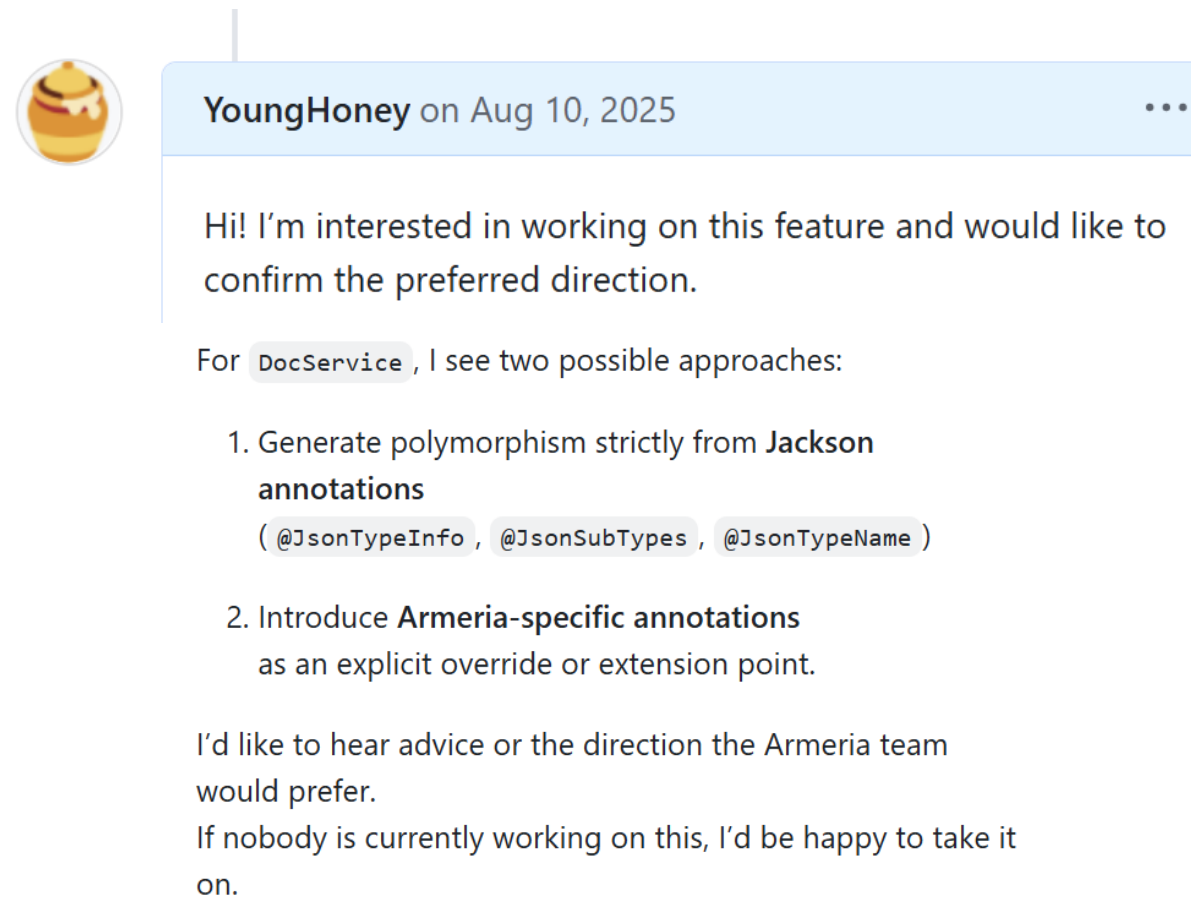
Structs

PolymorphismDocServiceDemo\$Animal
PolymorphismDocServiceDemo\$Zoo

■ 사용 기술 / 적용 맥락 및 이유

- Jackson Annotation

→ 다형성 문서화의 근거가 필요했고, Maintainer에게 문의한 결과 Jackson Annotation 방식을 선호한다고 응답받음



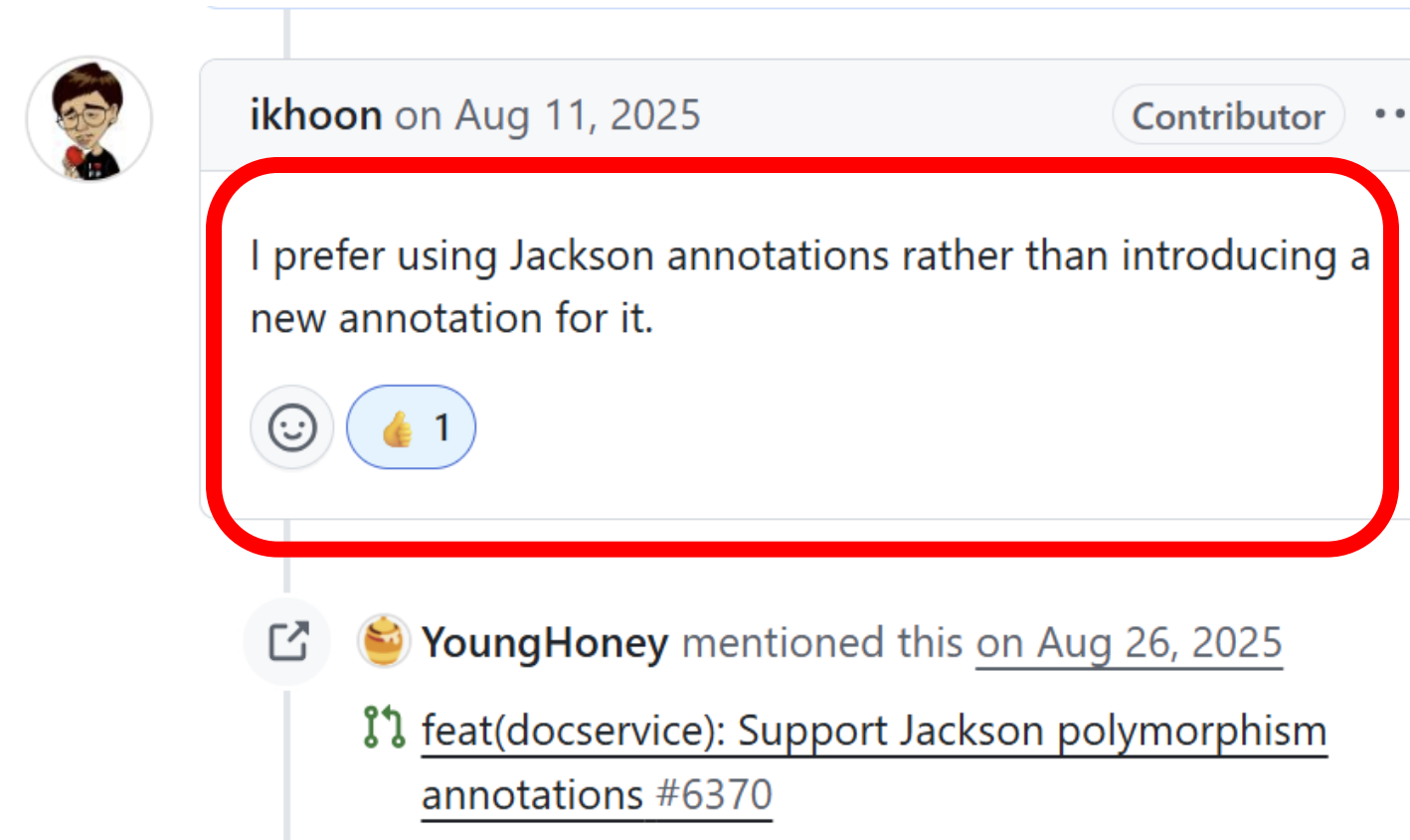
YoungHoney on Aug 10, 2025

Hi! I'm interested in working on this feature and would like to confirm the preferred direction.

For `DocService`, I see two possible approaches:

1. Generate polymorphism strictly from **Jackson annotations**
(`@JsonTypeInfo`, `@JsonSubTypes`, `@JsonTypeName`)
2. Introduce **Armeria-specific annotations** as an explicit override or extension point.

I'd like to hear advice or the direction the Armeria team would prefer.
If nobody is currently working on this, I'd be happy to take it on.



ikhoon on Aug 11, 2025 Contributor

I prefer using Jackson annotations rather than introducing a new annotation for it.

👍 1

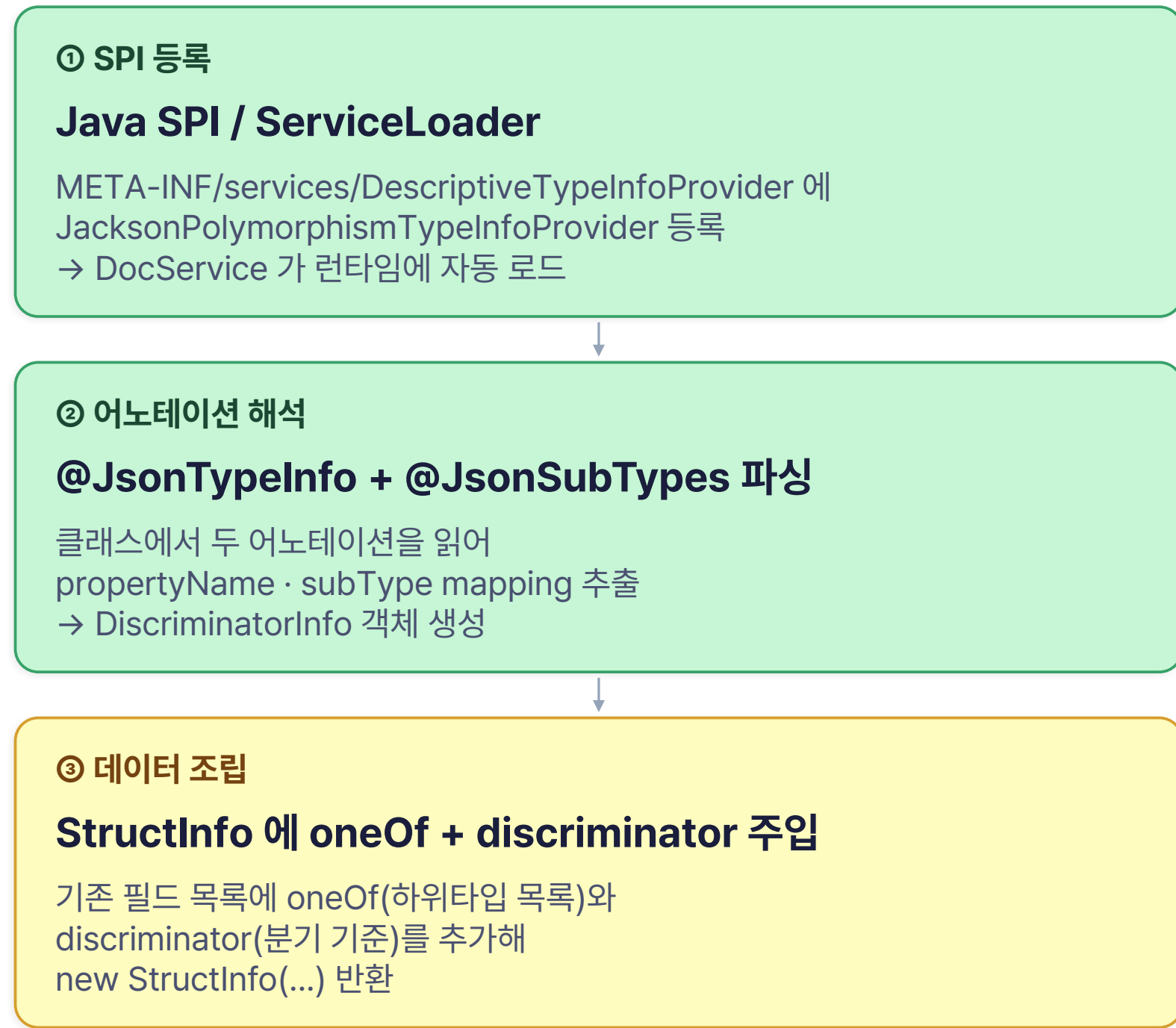
🔗 YoungHoney mentioned this on Aug 26, 2025

🔗 feat(docservice): Support Jackson polymorphism annotations #6370

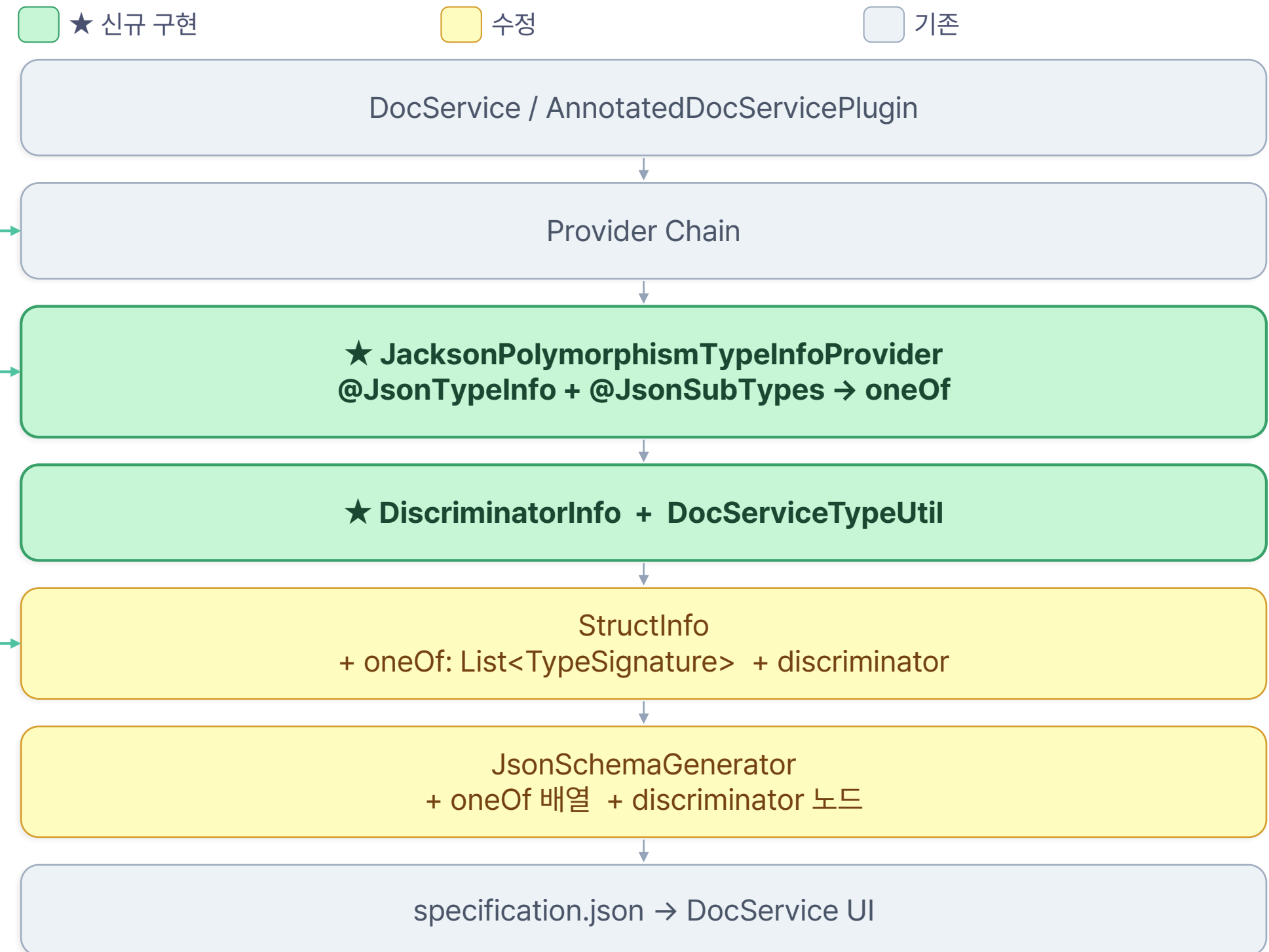
구현 — JacksonPolymorphismTypeInfoProvider

@JsonTypeInfo + @JsonSubTypes 를 읽어 oneOf + discriminator 를 생성하는 신규 Provider 구현

구현 내용



전체 파이프라인에서의 위치



결과

Armeria documentation service 1.37.1

Services

Overview

PolymorphismDocServiceDemo\$AnimalService ^

- POST** processAnimal()
exact:/api/animal
- POST** processAnimalRecord()
exact:/api/animal/record
- POST** processOptionalAnimal()
exact:/api/animal/optional
- POST** processZoo()
exact:/api/zoo

Structs

- PolymorphismDocServiceDemo\$Animal
- PolymorphismDocServiceDemo\$Zoo

^



Armeria documentation service 1.39.1 Go to ...

Services

Overview

PolymorphismDocServiceDemo\$AnimalService ^

- POST** processAnimal()
exact:/api/animal
- POST** processAnimalRecord()
exact:/api/animal/record
- POST** processOptionalAnimal()
exact:/api/animal/optional
- POST** processZoo()
exact:/api/zoo

Structs

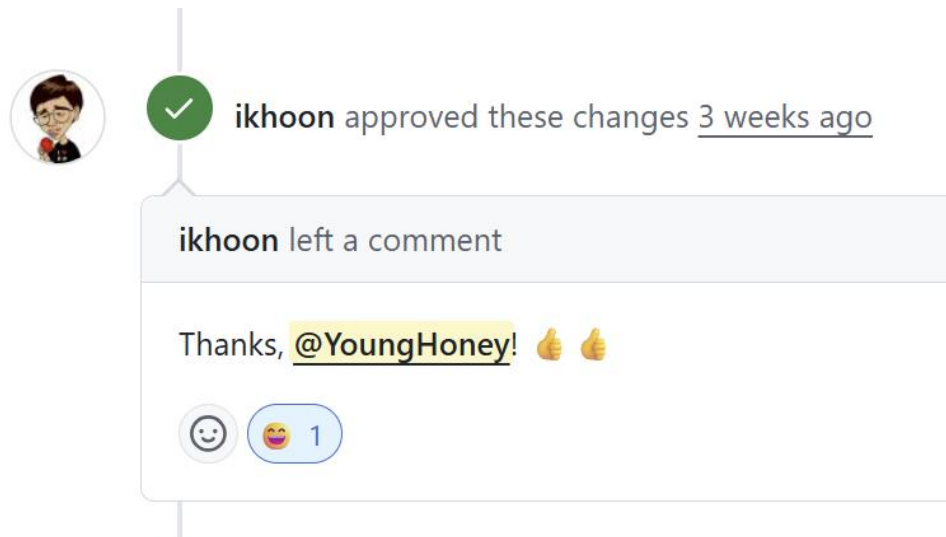
- PolymorphismDocServiceDemo\$Animal
- PolymorphismDocServiceDemo\$Cat
- PolymorphismDocServiceDemo\$Dog
- PolymorphismDocServiceDemo\$Toy
- PolymorphismDocServiceDemo\$VetRecord
- PolymorphismDocServiceDemo\$Zoo

PolymorphismDocServiceDemo\$Dog

com.linecorp.armeria.internal.server.annotation

Fields

Name	Required	Type
name	required	string
age	required	int
favoriteFoods	required	list<string>
favoriteToy	required	PolymorphismDocServiceDemo\$Toy
vetRecord	required	PolymorphismDocServiceDemo\$VetRecord



feat(docservice): Support Jackson polymorphism annotations #6370

Merged ikhoon merged 13 commits into line:main from YoungHoney:YoungHoney6313 6 hours ago

- `DocService` now correctly generates JSON Schema with `oneOf` and `discriminator` fields for types annotated with Jackson's `@JsonTypeInfo` and `@JsonSubTypes`. #6370

Thank you

This release was possible thanks to the following contributors who shared their brilliant ideas and awesome pull requests:



■ 프로젝트 성과

- 3명의 Maintainer에게 Approve를 받았으며, Merge되어 Contributor로서 등록 되었습니다.

■ 프로젝트 회고

- 오픈소스 기여에 관심이 컸지만, 감히 내가 해도 될까 하는 두려움이 있었습니다. 그러나 Maintainer들이 적극적으로 도와주셨고, 어느덧 작업을 마무리하게 되었습니다.

- 코드를 분석하면서 Mermaid로 클래스 다이어그램으로 나타내곤 했는데. DocService를 구성할 때 Builder패턴을 사용하고, Provider들을 사용하는 부분에서 Strategy 패턴과 CoR 패턴이 보였는데, 설계패턴 과목을 듣지 못했다면 알아채지 못했을 것 같습니다.

- 작업과정에서 null값 처리. 고려하지 못한 Case들에 대한 처리 등 실제 구현만큼 예외처리가 중요하다는 생각이 들었고, 실제로 Product를 생산하는 실무자로서 입력값에 대한 검증이 필수적이라고 생각했습니다.