

# 채기훈

## Backend Developer

읽기 쉬운 코드와 검증 가능한 구조를 통해 팀이 빠르게 이해하고 안전하게 수정할 수 있는 시스템을 만드는 2년차 백엔드 개발자입니다.



chae0738@naver.com

010-6278-8593

github.com/Hun425

velog.io/@chae0738

#단순하고 명확한 설계 — 오버엔지니어링을 지양하고 의도와 책임이 분명한 구조를 지향

#의도가 드러나는 코드 — 흐름과 의도가 자연스럽게 읽히는 코드로 가독성과 협업 효율 향상

#AI 활용 능력 — AI Agent를 학습·구현·정리에 적극 활용해 반복 업무 절감

## 기술 스택

**Backend** Kotlin · Java · Spring Boot · Armeria · gRPC · Redis · Kafka

**Database** PostgreSQL · MySQL · Elasticsearch

**DevOps** AWS · Docker · Linux · Jenkins

## 경력

**비온드메디슨** 엔터프라이즈개발팀 · 팀원 2026.03 - 재직중

**Clickless** Kotlin · Spring Boot · Armeria · gRPC · Docker · MySQL · Redis

- hexagon 아키텍처 기반 포트/어댑터 계층 분리 설계, 비즈니스 규칙과 인프라 의존성을 격리하여 유지보수성 확보
- 내부 서비스 간 계약 안정성과 인증·세션 컨텍스트 일관성을 위해 gRPC + Armeria 기반 API 개발
- 테스트 구조 최적화를 통해 전체 테스트 실행 시간 **약 88% 단축** (429초 → 50초)

**마드라스체크** 엔터프라이즈개발팀 · 팀원 2025.10 - 2026.03

**삼성화재 통계 고도화** Java · Node.js · PostgreSQL · Redis

- 기존 WAS에 직접 집계 로직을 넣지 않고, Java 백엔드-Node.js-Redis-PostgreSQL로 이어지는 분리형 구조를 설계해 실시간 이벤트 수집과 배치 이관 분리
- 사용자 액션 기반 이벤트 발행, 배치 Worker, 실시간 마이그레이션, 실패 시 로컬 백업까지 고려한 집계 파이프라인 구현

**아이씨티웨이** SI 2팀 · 사원 2024.12 - 2025.10

**대기정책 시스템 구축** Java · Spring Boot · Kafka · Spark · PostgreSQL · MyBatis

- Kafka와 Spark 기반 분산 처리 아키텍처를 설계하여 대용량 차량 데이터 처리 파이프라인 구축
- 비동기 인덱스 생성 전략 도입으로 데이터 적재 성능 **60% 개선** (1시간 40분 → 40분)
- 배치 처리 구조 개선으로 전체 처리 시간 8~10일 → 3~4일, **약 60% 단축**
- Apache POI 기반 데이터 파싱 및 적재 로직 설계로 안정적인 데이터 처리 환경 구축

**현대자동차 PMS 설계** PostgreSQL · Data Modeling · SQL

- 역할분류와 TASK 분류를 동일 구조로 단순 통합하지 않고, 도메인 규칙과 변경 책임의 차이에 따라 관리 정책과 데이터 모델을 분리 설계
- 프로젝트별 독립 관리가 필요한 영역과 공통 카탈로그로 관리해야 하는 영역을 구분하고, 컨텍스트에 맞는 복사/공유 구조와 계층형 분류 모델을 정의

## 주요 과제 & 성과

## 테스트 구조 최적화를 통한 CI 피드백 루프 개선

비온드메디슨 · 성능 개선

### 배경

Clickless 저장소의 통합 테스트가 @SpringBootTest + Testcontainers + @DirtiesContext 조합으로 인해 컨텍스트 재기동이 반복 발생. 전체 테스트 실행에 429초 소요.

### 해결

병목이 테스트 로직이 아닌 컨텍스트 재기동 비용임을 baseline 측정으로 확인. 테스트 의미를 보존하면서 @DirtiesContext 대신 Redis cleanup + DB snapshot restore로 전환하여 컨텍스트 캐시 재사용을 극대화하는 방향을 선택.

### 결과

전체 테스트 **429s → 50s (88% 단축)**  
최대 병목 **207s → 3s (98% 단축)**

## Elasticsearch 도입을 통한 대용량 데이터 검색 성능 최적화

아이씨티웨이 · 성능 개선

### 배경

약 2,260만 건의 차량 등록 데이터 검색 시 PostgreSQL LIKE 쿼리로 인한 Full Table Scan 발생. 마지막 페이지 조회 시 응답 시간 최대 20초까지 소요.

### 해결

GIN 인덱스 전문 검색도 검토했으나 OFFSET 구조적 한계 해결이 불가하다고 판단. 역색인 기반으로 검색 접근 방식 자체를 전환하기 위해 Elasticsearch를 선택하고, 데이터 정합성이 중요한 원본은 PostgreSQL에 유지하는 하이브리드 구조 설계.

### 결과

검색 응답 **20s → 2s (90% 단축)**  
DB 부하 **70% 감소**

## Caffeine 캐시 적용을 통한 퀴즈 API 성능 최적화

팀 프로젝트 · 성능 개선

### 배경

퀴즈 목록/상세 조회 API의 반복적 DB 접근으로 인한 성능 저하. 특히 인기 퀴즈나 데일리 퀴즈의 경우 불필요한 DB 부하 발생.

### 해결

조회 빈도는 높지만 변경 빈도가 낮은 데이터 특성을 분석하여 분산 캐시 대신 로컬 캐시가 적합하다고 판단. Caffeine을 선택하고, 데이터 변경 시 @CacheEvict로 일관성을 보장하는 전략 적용.

### 결과

응답 속도 **70% 단축**

## 교육 & 자격

### 교육

2024.01 - 2024.12

삼성 청년 SW 아카데미 (SSAFY) 11기 · 웹 기술 트랙 수료

2017.03 - 2023.08

경상국립대학교 · 도시공학과 전공 / 스마트도시건설 복수전공

### 자격증 & 수상

정보처리기사 한국산업인력공단

삼성 역량 평가 PRO (B형) 삼성전자 2024.06

공동 프로젝트 우수상 삼성전자 2024.08