



전수민 백엔드 개발자

- ✉ wjsaos2081@gmail.com
- 🌐 <https://dalcheonroadhead.tistory.com>
- 🔗 <https://github.com/dalcheonroadhead>

INTRODUCE

1. 기능 구현 후 3회 이상 성능 점검

- 부하, 시나리오, 스파이크 테스트 진행 → TPS, RPT, 에러 수집, 병목 지점 확인
- Core Service, 최소 3개 이상의 Edge Case 테스트 코드 작성 및 확인

2. 병목 원인, 예측보단 지표 기반 최소 2회 이상의 다른 접근

- 성능 지표 개선될 때까지, OS, WAS, query, caching 등 다각적으로 방법 적용

3. 특정 기술에 구애받지 않음.

- 첫 기능 구현은 무조건 spring 자체 기능과 Legacy 라이브러리 활용
- 회사 기술 stack에 빠르게 적응할 준비 되어있음.

TECH STACK

Backend

- Java
- Spring Boot
- Spring MVC
- Spring Security
- Junit
- JPA
- Querydsl
- JDBC
- MyBatis
- Feign Client

DevOps

- MySQL
- Redis
- Kafka
- Jmeter
- Grafana
- Tomcat
- AWS Lambda
- Cloud Watch

Frontend

- React.js
- vite
- Npm

PROJECT

SPOT

2025.02 ~ 2025.04

[https://github.com/6-](https://github.com/6-SPOT/spot-be)

SPOT/spot-be

개요: 하기싫은 소일거리 매칭 서비스 ('당근알바', '해주세요' 클론 코딩)

담당: 구인구직 매칭 서비스 개발

역량:

사용자 위치 기반 반경 N km 내 일거리 찾기 API 개선

(RPT 3102ms → 170ms, TPS: 50 → 270)

- v-user 3000명, 일거리, 회원 데이터 20만 개, 1000RPS, 7분 부하 지속
- [쿼리 전략 최적화\(JPQL, native-query, quertdsl 부하 테스트 및 적용\)](#)
- [강제 힌트 적용, 커버링 인덱스, 쿼리 튜닝으로 쿼리 속도 개선\(1.2초 → 0.102초\)](#)
- 데이터 분산 및 밀집 분포 **30회** 달리 설정하며, 옵티마이저 실행계획 분석
- [R-Tree, Spatial Index, 내장 공간 함수 적용 및 개선 \(3102ms → 170ms\)](#)
- 부하 테스트 **30회** 시행 및 **4번**의 고도화 완성
- 기술 보고서: <https://github.com/dalcheonroadhead/spot-be>
- 발표 자료: <https://cs-study-soomin.my.canva.site/job-searching>

구직자 NO-SHOW 시 일 매칭 10분 예약 철회 API 개선

(대용량 데이터 처리 300RPS 오류율 25.74% → 3000RPS 오류율 0.0%)

- v-user 5000명, 매칭 API 5개 + 예약 철회 시나리오, 300RPS 부터 30초마다 300 RPS 씩 점진적 부하 5분 30초 지속
- 동기 스케줄러 → 비동기 스케줄러 + 대기큐 고도화 (타 API 부하 부담 0% 줄임)
- [1800RPS에서 쓰레드 업무 거절 예외 → 지수 백오프 전략 재시도 도입](#)
- [도입 후 오히려 퍼포먼스 저하\(→ 1200PPS\) → 지수 변이 추가 지수 백오프 전략 구현 \(1200RPS → 3000RPS 개선\)](#)
- 16회 시나리오 테스트 보고서: <https://github.com/6-SPOT/spot-be/wiki>

팀 테스트 환경 구성 (WAS, OS 튜닝, Cloud Watch SQL, 서버 에러 로그 확인)

- API 테스트 시, 테스트 시작인 300RPS도 못 버티고 30초만에 다운, yaml 파일 내 tomcat 설정 변경 해도 차도 없음
- OS 커널 내 SYN 큐와 ATK 큐 사이즈, NIC 연결 큐 사이즈 확장 (300 RPS → 1800RPS)
- 트러블 슈팅: <https://github.com/6-SPOT/spot-be/discussions/274>
- 슬로우 쿼리 설정, Cloud Watch 설정으로 에러 로그 수집 및 쿼리로 찾도록 구현

PROJECT

WALK-WALK

2024.02 ~ 2024.04

<https://github.com/dalcheonroadhead/walk-walk>

개요: 할아버지, 할머니 조강하신 거리만큼 용돈 드리는 서비스

담당: STT,TTS 채팅 서비스 (음성 → 채팅 메시지 변환, 메시지 → 음성 변환)

역량:

실시간 **STT (음성 → 채팅 메시지) 변환 아키텍처 개선**
(RPT 20s→ 7s)

- v-user 500, 500RPS, 3분 지속
- AWS Transcribe(STT 서비스)에 요청 및 응답 동기적 대기 초기 구현(20초)
- 500 RPS 부하 시, 동기 요청으로 인한 Thread TimeOut, AWS 요청 파일이 다른 요청 파일에 의해 덮어써지는 오류 발생
- AWS Lambda 활용 (AWS Transcribe → S3 파일 자동 저장 함수 구현)
- 백그라운드 쓰레드가 HTTP 폴링으로 S3에서 올바른 파일 가져오도록 구현
- 오류율 32.74% → 0.0%, RPT: 20s → 7s 개선
- 기술 보고서: [링크](#)

EDUCATION

PROPECT

2025.02 ~ 2025.04

주최: (주) groom

개요: Back-end 개발자 단기 심화 과정

역량:

- 정확한 지표 확인법 학습 및 성능 개선 방향성 제시가 필요해 참여
- 성능 테스트 툴 (Jmeter, k6, Grafana) 활용법 및 지표 확인 방법 학습

삼성 청년 소프트웨어 아카데미

2023.07 ~ 2024.06

주최: (주) 삼성 전자

개요: 전공자 백엔드 개발자 과정 1년 **우수 수료 (상위 30%)**

역량:

- 프로젝트 경험 부족 및 백엔드 개발 기본 역량 부족하여 참여
- Spring Boot, JPA, MySQL 등 백엔드 개발 기본 역량 증진
- **4회 프로젝트 4회 수상**
- 이 달의 멤버상 **2회** 수상

홍익대학교

2017.03 ~ 2023.08

개요: 법학과(주), 컴퓨터공학과 (복수 전공)

역량:

- 개발 기초 역량 학습을 위해 복수전공 이수 및 수료
- SW / 데이터 활용 역량 인증 취득

Experience

real my sql study

2025.02 ~ 현재 진행 중
<https://github.com/RealMySQL-Gang/real-my-sql-study>

개요: real-my-sql 책 요약 및 내용 실습 스터디

역량:

- MVCC, 격리 수준 4 단계, 버퍼 풀 등 mySQL 내부 아키텍처와 원리 이해의 필요성을 느끼고 개설
- 매주 수요일마다 발표 진행 중

구름톤

2024.10 ~ 2024.11
주최: kakao x groom
<https://github.com/dalcheonroadhead/groomthon-11th-server>

개요: 백엔드 개발자 5인 중 한 명으로 선발 및 3박 4일 과정 수료

역량:

- 인프라 역량 증진 필요성을 느끼고 참여
- 크래프린과 kakao 내부 Docker 서버를 활용해 서버 배포 경험
- Nat-gateway와 private subnet 이론과 활용법 학습

새싹톤

2024.8 ~ 2024.9
주최: 서울 경제 진흥원 x 구글
<https://github.com/sesac-dev/eazione-back>

개요: 50팀 중 6등으로 본선 진출 및 2주 과정 수료

역량:

- 기한 내 MVP 개발 역량을 쌓기 위해 참여
- 처음 조우한 디자이너 및 PM과 협업 역량 증진

신한 해커톤

2023.9 ~ 2023.9
주최: 서울 경제 진흥원 x 구글
<https://github.com/dalcheonroadhead/solsol-pokect>

개요: 100팀 중 25팀에 선발 및 1박 2일 연수원 본선 과정 수료

역량:

- 신한은행 내부 계좌 이체 API 활용을 경험할 수 있어 참여
- 현직자 멘토링을 통해, 현직 수준의 성능 테스트와 개발 역량을 확인

Open source

채팅 서비스

쉽게 따라하기

2024.04 ~ 2024.05
server:
<https://github.com/dalcheonroadhead/real-time-chat-server>
client:
<https://github.com/dalcheonroadhead/real-time-chat-client>

개요: 실시간 이미지 업로드도 가능한 STOMP, SPRING 기반 채팅 서비스

역량:

- 이미지 업로드가 가능한 채팅 서비스 구현에 많은 동기들이 어려움을 겪고 있는 것을 보고 견본용 프로젝트 구현
- STAR 총합 : 19 개 (20250414 기준)