

이건호

Lee GeonHo

Contact

010-3376-9792

thisishosupply@gmail.com

Channel

<https://github.com/This2sho>

<https://this2sho.github.io/>

About Me

- 저는 팀워크와 성장을 중시하는 개발자입니다.
- 최고의 결과를 달성하기 위해 명확한 소통을 선호하며, 끊임없는 배움에 열정을 가지고 있습니다.
- 코드 리뷰와 건설적인 피드백을 통해 팀과 함께 성장하는 것을 목표로 하고 있습니다.

Stack

Back-End

- Java
- Spring Boot, Spring MVC, Spring Data JPA
- JPA, Querydsl
- Junit5, Mockito

DevOps

- AWS EC2, S3, RDS, CloudFront
- Docker
- Mysql 8, flyway
- Jenkins, Github actions

Collaboration

- Github, Git Projects
- Slack, Notion

주차의 상상은 현실이 된다 (2023.12~present)

목적지 근처 주차장을 추천하는 웹 애플리케이션
<https://github.com/parkingkim/parking-BE>

담당 작업

(1) 주차장 데이터 스케줄러 개발

- 고민했던 부분

실시간으로 주차장 정보를 제공하는 시스템에서 **네트워크 통신문제와 성능 최적화**가 가장 큰 고민이었습니다.

주차장 정보가 실시간으로 갱신되어야 했으나, 외부 API와의 통신이 간헐적으로 실패하면서 주차장 데이터를 받아오지 못하는 문제가 발생했고, 이를 어떻게 해결할지 많은 고민을 했습니다.

- 해결 과정

네트워크 통신 문제

문제 해결을 위해 **외부 API가 정상적으로 작동하는지 사전 확인하는 예비 요청**을 구현했습니다.

예비 요청이 성공했을 때만 데이터를 받아오도록 하여 불필요한 리소스 소비를 방지했습니다.

네트워크 요청 실패 시 **재시도 로직**을 추가하여 간헐적인 통신 실패 문제를 해결했습니다.

또한, **요청 실패율이 특정 수치를 넘어서면 일정 시간 요청을 중단 후, 다시 요청**을 수행하도록 조치하였습니다.

성능 최적화

데이터 업데이트 과정에서 비동기 처리를 도입하여 **네트워크 I/O 시간을 줄였고, 전체 처리 시간을 약 85% 개선**했습니다. 또한, 데이터를 저장하는 과정에서 개별 쿼리로 데이터를 저장하던 로직을 batchUpdate로 수행하도록 하여, 13,444 데이터 기준 **삽입 시간을 16초에서 4초로 75% 개선**했습니다.

- 얻은 것

이 작업을 통해 외부 시스템에 장애가 있을 때, 어떻게 하면 서비스에 미치는 영향을 최소화하고 회복력을 높이는 방법을 이해하게 되었습니다. 또한 네트워크 문제를 해결하기 위해 여러 시도를 해보면서, 다양한 접근 방법의 중요성을 배울 수 있었습니다.

(2) CI/CD 인프라 구성

- 고민했던 부분

초기 서울을 기준으로 주차장의 실시간 주차대수를 제공하고 운영하면서 계속해서 여러 지역의 주차장 정보를 추가하고자 했습니다. 이 과정에서 **다른 지역의 주차장 정보를 추가하면서 서비스는 중지되지 않기**를 원했습니다. 기존 시스템은 다른 주차장을 추가하여 배포할 때마다 전체 시스템이 중단되었습니다.

- 해결 과정

이를 해결하기 위해 **도메인 모듈, API 모듈, 스케줄러 모듈을 분리**하여 각각 독립적으로 배포가 가능한 방향을 설계했습니다. 이후, 배포 파이프라인을 수정하여 변경된 모듈만 자동으로 테스트하고 배포되도록 구현했습니다. 이를 통해 기존 서비스하는 API 모듈은 중단되지 않고 주차장 데이터(스케줄러 모듈)를 업데이트 할 수 있게 되었습니다.

- 얻은 것

이 작업을 통해 멀티 모듈의 장점을 직접 체감할 수 있었으며, 자동화된 배포 프로세스를 통해 개발 생산성이 크게 향상되었습니다.

(3) 개선 작업

- 고민했던 부분

프로젝트 초기에 엔티티 간의 의존성이 강하게 결합되어 있어, 테스트 환경에서 객체 생성을 어렵게 만들고, 코드 변화에 유연하지 못했습니다. 추가로, 제일 많이 사용하는 주차장 조회 API에서 풀테이블 스캔을 사용하여 수정할 필요가 있다고 생각했습니다.

- 해결 과정

엔티티 강결합

기존 객체 참조 대신 ID 참조로 변경하여 엔티티 간 의존성을 줄였고, 그로 인해 의존 관계가 명확하지 않은 부분에 대해서는 값 객체와 제네릭을 사용하여 의존 관계를 명확하게 파악할 수 있도록 변경하였습니다. 이를 통해 테스트 코드 작성 시, 객체 생성을 간소화하고 코드 변경에 대응하기 쉬워졌습니다.

풀테이블 스캔

주차장 API에서 발생하는 풀 테이블 스캔 문제를 해결하기 위해, 공간 인덱스를 추가하여 인덱스 범위 스캔을 가능하게 하였습니다. 이를 통해 50명의 사용자가 목적지 반경 500m 이내의 주차장을 조회할 때 평균 응답 시간이 3.85초에서 0.16초로 약 95% 개선되었습니다.

- 얻은 것

객체간 강결합의 문제점과 코드 변경시 발생할 수 있는 문제점들을 경험했습니다. 또한, 객체 참조 구조를 개선함으로써 가독성 좋은 코드에 대해 고민하게 되었습니다. 쿼리 분석과 API 성능 측정을 하면서 DB 인덱스에 대해 학습할 수 있었습니다.

Project

이돈이면

(2023.06~2023.12)

사용자들에게 소비 절제와 합리적인 소비를 도모하며
재미와 유용한 정보를 제공하는 안드로이드 어플리케이션
<https://github.com/woowacourse-teams/2023-edonymyeon>

담당 작업

- 비밀번호 암호화 기능 구현

어플리케이션 배포 전 개인정보처리방침에 따라 암호화 기능 필요하여, salt를 적용한 비밀번호 해싱 기법을 도입하여 비밀번호를 암호화하도록 구현. salt는 데이터베이스에 저장하지 않고 애플리케이션 단에서 관리하여 해시값과 함께 비밀번호를 검증하는 방식으로 보안 강화.

- 인증 기능 구현

초기 Basic 인증 방식에서 Session 방식으로 변경하여 인증 기능 구현 및 사용자 편의성을 고려하여 소셜 로그인 기능 구현. 이후, 서버 잦은 서버 업데이트로 세션 정보가 유실되어 사용자 불편 발생하여 Redis를 도입해서 서버 업데이트 후에도 세션 유지가 가능하도록 개선.

- 알림 기능을 사용하는 API에서 타임아웃 문제 해결

기존 댓글 작성, 게시글 좋아요 기능등을 사용하면 게시글 작성자에게 알림을 전달하는데, 부하 상황에서 스레드간 커넥션을 차지하기 위한 레이스 컨디션이 발생. 이를 해결하기 위해, 알림이 발생하는 행동(댓글 작성, 좋아요)와 알림을 비동기로 처리하여 동시에 커넥션을 하나만 사용하도록 개선.

Activites

우아한테크코스 5기 백엔드 (2023.02 ~ 2023.11)

우아한테크코스 활동 기록

<https://github.com/This2sho/Woowacourse>

사다리, 체스 등 미니게임을 개발하며 객체 지향 프로그래밍, 스프링, 테스트 코드에 대한 학습

- tomcat, spring mvc, jdbc 라이브러리 개발을 통한 핵심 라이브러리 이해
 - 레거시 코드 리팩토링을 통한 유지보수
 - 페어 프로그래밍과 코드 리뷰 문화 경험을 바탕으로 프로젝트에 직접 적용
 - 부족한 부분 스터디를 통한 학습 (이펙티브 자바, 성공과 실패를 결정하는 1%의 네트워크 원리)
 - 모던 자바에 대해 학습하고 해당 주제로 테코톡 발표 진행
-

Education

부산대학교 컴퓨터 공학과 (2016.03 ~ 2022.08)

교내 앱 개발 동아리 활동

- 기획자, 디자이너들과 소통하며 안드로이드 앱 개발 및 플레이스토어에 앱 출시
- 제 4회 창의융합 소프트웨어 해커톤 대회 장려상 (2021.06 ~ 2021.09)

농구 동아리 활동

- 부산대학교 전기,전자, 컴퓨터 농구 동아리 임원 경험
 - 신입 부원 회비 무료 및 홍보물 제작으로 신입 회원 전년도 대비 약 2배 증가
 - 회비 사용 내역 작성 및 동아리 MT 일정 문서화
 - 코로나로 인한 교내 대회 금지, 농구 코트 사용불가로 연습 공간이 부족했지만 농구 코트 대관 및 연휴를 이용한 졸업한 선배들과 기수 별 대회 개최
-

Etc.

AWS certified Solutions Architect Associate

- AWS 기술 기반으로 가용성, 확장성을 갖춘 시스템 설계 학습
- 2022.08.25 취득

정보 처리 기사

- 2022.06.17 취득