

# 심준보 | Backend Engineer

Java · Spring · SQL · 운영 자동화

약 690만 회원 규모의 책이음 전국 도서관 연계 서비스에서 통계 성능, 데이터 정합성, 운영 조회 흐름을 개선해 왔습니다. 문제의 원인을 데이터와 업무 흐름에서 찾고, 서비스 운영에 필요한 기능을 구현하는 데 집중했습니다.

## 사례

| 사례        | 문제와 담당 범위              | 핵심 내용            |
|-----------|------------------------|------------------|
| 01 통계 조회  | 통계 테이블·집계/조회 쿼리·인덱스 설계 | 약 30분 → 100ms 이하 |
| 02 회원수 수집 | 200개 이상 지역의 수동 수집 자동화  | 그룹 분할과 병렬 처리     |
| 03 데이터 보정 | 회원 상태·날짜 누락별 보정        | 유효한 기존 값 유지      |
| 04 이기종 조회 | 설정별 조회 경로와 응답 변환       | 운영 화면에서 직접 조회    |
| 05 민원 조회  | 기본정보·가입지역·연체자료 통합      | API·SQL·화면 개선    |

### 기술 스택

Java, Spring Framework, Spring Boot, Spring MVC, MyBatis, JSP, Ajax, Oracle-Tibero SQL, Netty, MQTT, Quartz, JMeter, Git

# 대용량 통계 조회 성능 개선

(주)이씨오 | 책임음 개선 사업 | Oracle · Tibero

이용자별 독서순위·도서별 대출순위 조회를 위해 통계 테이블 설계, 집계·조회 쿼리 개발, 인덱스 설계와 튜닝을 담당했습니다.

## 문제와 접근

통계 조회에 약 30분이 걸려 결과를 즉시 확인하기 어려웠습니다. 실행계획을 분석하고 테이블 구조·인덱스·조회 쿼리를 개선했습니다. 대출 데이터를 배치에서 집계하고, 화면은 통계 테이블을 조회하도록 구성했습니다.

|    |                                    |
|----|------------------------------------|
| 01 | <b>원천 대출 데이터</b><br>대출현황·상호대차·대출이력 |
| 02 | <b>기간·회원별 집계</b><br>배치에서 통계 계산     |
| 03 | <b>통계 테이블</b><br>회원·연도 기준 갱신·삽입    |
| 04 | <b>화면 조회</b><br>순위·평균·대출량 반환       |

| 판단        | 구현                   | 고려한 점                  |
|-----------|----------------------|------------------------|
| 집계와 조회 분리 | 배치 집계 결과를 통계 테이블에 저장 | 원천 데이터 집계를 화면 요청에서 분리  |
| 집계 기준     | 회원 키와 연도를 기준으로 갱신·삽입 | 회원별 연간 대출량을 같은 기준으로 관리 |
| 순위 계산     | 전체 순위 계산 후 대상 회원 조회  | 동점자는 RANK로 같은 순위 부여    |

**프로젝트 결과**  
테이블 구조·인덱스·쿼리 개선을 통해 통계 조회 시간을 약 30분에서 100ms 이하로 단축했습니다.

# 지역별 회원수 수집 자동화

㈜이씨오 | 2025.11 | 스케줄러 · 병렬 처리

담당자가 수동으로 하던 200개 이상 지역의 회원수 수집을 스케줄러 기반으로 자동화했습니다. 작업 분할과 지역별 조회·검증·저장 흐름을 개발했습니다.



작업 그룹의 실행 흐름을 나타낸 개념도입니다. 처리 시간이나 고정 스레드 수를 나타내지 않습니다.

## 작업을 20개 단위로 나눈 이유

- 전체 지역을 동시에 호출할 때의 부하와 한 지역씩 호출할 때의 대기 누적을 함께 고려했습니다.
- 지역 목록을 최대 20개씩 묶고 그룹 안에서 `parallelStream`으로 작업을 분배했습니다.
- 그룹의 마지막 작업이 반환돼야 다음 그룹을 시작하므로, 느린 지역의 대기도 해당 그룹의 진행에 영향을 줍니다.

# 그룹·스레드·트랜잭션의 경계

그룹 간 순차 실행 · 그룹 안에서 병렬 실행

| 단위     | 동작                                                                 |
|--------|--------------------------------------------------------------------|
| 작업 그룹  | 최대 20개 지역. 현재 그룹의 모든 작업이 반환된 뒤 다음 그룹 실행                            |
| 실행 스레드 | parallelStream의 공용 ForkJoinPool과 호출 스레드 사용. 20은 고정 스레드 수가 아닌 그룹 크기 |
| 요청 데이터 | 프레임 객체와 조회·저장용 Map을 지역 작업별로 분리                                     |
| 대기·예외  | 연결 읽기 타임아웃은 각각 5초. 지역 작업의 예외를 기록하고 다른 지역 처리를 이어감                   |

## 지역별 서비스 호출

주입된 서비스 프록시를 호출해 대표도서관 확인·원격 회원수 조회·월별 저장을 수행합니다. 이 서비스 호출에는 기존 Spring REQUIRED 규칙이 적용되며 원격 호출 대기가 포함될 수 있습니다. 20개 지역 그룹 전체를 하나의 트랜잭션으로 묶지는 않습니다.

## 지역별 요청을 만들고 서비스 프록시 호출

```
for (List<String> group : partition(regions, 20)) {
    group.parallelStream().forEach(region -> {
        try {
            Frame request = newRequest(region);
            statisticsService.collectAndStore(request);
        } catch (Exception e) {
            logFailure(region, e);
        }
    });
}
```

설명용 제어 흐름 · 서비스명과 업무 식별자는 일반화했습니다.

# 회원 상태에 따른 데이터 보정

(주)이씨오 | 책이음 연계 서비스 | Java · 업무 규칙

지역별 동기화 오류를 추적하고 회원 상태와 날짜 누락 여부에 따라 대출정지일 보정 규칙을 나눴습니다. 누락된 값으로 센터의 유효한 정지 정보가 지워지는 상황을 고려했습니다.

| 수신 상태       | 보정 후보           | 처리 이유                      |
|-------------|-----------------|----------------------------|
| 정상 회원       | 대출정지일을 빈 값으로 변경 | 정상 상태에 남은 정지일 정리           |
| 정지 회원·날짜 누락 | 기존 센터 정지일 유지    | 누락 값으로 유효한 정지일이 지워지는 상황 방지 |
| 정지 회원·날짜 존재 | 도서관의 정지일 사용     | 업무 상태에 맞는 값을 비교 후보로 선택     |

## 보정 흐름

|    |                                |
|----|--------------------------------|
| 01 | <b>도서관 값 수신</b><br>회원 상태·대출정지일 |
| 02 | <b>보정 후보 결정</b><br>정상·정지·누락 구분 |
| 03 | <b>날짜 비교</b><br>센터 값의 일자와 대조   |
| 04 | <b>필요한 값 갱신</b><br>차이가 있을 때 보정 |

## 연관 개선 | WEB 탈퇴 처리 일관성

WEB 탈퇴 경로의 처리를 팀에서 합의한 기존 soft delete 정책에 맞춰 수정했습니다. 탈퇴 지역의 회원 상태를 변경하고, 탈퇴하지 않은 가입지역이 더 이상 없을 때 대상 개인정보 필드를 NULL로 정리했습니다.

# 이기종 도서관 이용자 직접 조회

㈜이씨오 | 2026.06 | Spring MVC · MyBatis

운영 화면에서 조회하기 어려웠던 이기종 도서관의 이용자 직접 조회 기능을 개발하고 운영에 반영했습니다.

## 문제와 담당 범위

기존에는 도서관 유지보수업체·사서에게 확인하거나 별도 쿼리를 수동 요청해야 했습니다. 도서관 설정에 따라 조회 경로를 선택하는 분기와 시스템별 응답을 공통 이용자 정보로 변환하는 처리를 개발했습니다.

|    |                                    |
|----|------------------------------------|
| 01 | <b>도서관 설정 확인</b><br>유형·별도 조회 설정    |
| 02 | <b>조회 경로 선택</b><br>기존 경로 또는 이기종 경로 |
| 03 | <b>응답 검증·변환</b><br>시스템별 항목 해석      |
| 04 | <b>공통 정보 표시</b><br>이용자 상태·대출정지일 등  |

| 판단        | 구현                          | 운영에서의 의미             |
|-----------|-----------------------------|----------------------|
| 설정에 따른 분기 | 별도 조회 설정이 있는 이기종 도서관에 분기 적용 | 기존 조회 경로를 유지하며 범위 확장 |
| 응답 구조 통일  | 서로 다른 응답을 공통 이용자 항목으로 매핑    | 시스템 차이를 화면에서 일관되게 표시 |
| 비정상 응답 구분 | 오류 응답·파싱 실패·이용자 정보 누락 확인    | 오류를 정상 조회 결과와 구분     |

## 결과

외부 담당자 확인이나 수동 쿼리 요청을 거치던 이용자 조회를 운영 화면에서 수행할 수 있게 했습니다.

# 주말민원 통합 조회 도구

㈜이씨오 | Spring MVC · MyBatis · JSP · AJAX

기본정보 중심이던 화면을 확장해 가입지역 상태와 연체자료를 함께 확인하도록 API·SQL·화면을 개선했습니다.

## 기본정보·가입지역 상태·연체자료 통합 조회

기능 비교용 재구성 화면 · 가상 데이터

### 책이음 이용자 동기화

DEMO-0001 (가상 이용자)

이용자 조회

동기화 실행

#### 이용자 정보

|       |           |    |        |     |         |
|-------|-----------|----|--------|-----|---------|
| 대출자번호 | DEMO-0001 | 이름 | 예시 이용자 | 연락처 | 연락처 비공개 |
|-------|-----------|----|--------|-----|---------|

#### 가입지역 상태

| No | 가입지역    | 회원상태 | 대출중 | 대출정지일      |
|----|---------|------|-----|------------|
| 1  | 예시 지역 A | 정상   | 0   | -          |
| 2  | 예시 지역 B | 정지   | 0   | 2026-10-15 |
| 3  | 예시 지역 C | 정상   | 0   | -          |

#### 연체자료 목록

| No          | 구분 | 도서관 | 서명 | 반납예정일 | 연체일수 |
|-------------|----|-----|----|-------|------|
| 연체자료가 없습니다. |    |     |    |       |      |

#### 이용 방법

- 이용자 번호를 입력하여 이용자 조회를 합니다.
- 조회된 가입지역과 연체자료를 확인합니다.
- 동기화 요청 후 상태를 다시 확인합니다.

가상 데이터로 재구성한 확장 화면의 조회 후 상태입니다.

- 조회 성공 전에는 동기화 버튼을 비활성화하고, 요청 중 버튼 상태를 관리했습니다.
- 동기화 완료 응답 후 다시 조회해 화면에 결과를 반영했습니다.

# 통신 서버 개발과 개인 프로젝트

이전 IoT 실무 · 교육 프로젝트 · 개인 작업

## IoT 실무 | (주)에이엘에이엔

Netty 기반 LTE 기기 TCP 서버에서 패킷 인코딩·디코딩, CRC-16, 암호화와 FOTA 기능을 개발했습니다. 통신 테스트용 클라이언트·서버를 구현하고 홈 IoT WAS의 N:M 관계 테이블, 센서 데이터 수집·처리, 디바이스 제어 API, MQTT 연동, Quartz 스케줄링을 개발했으며 JMeter 테스트를 진행했습니다.

## 교육 팀 프로젝트 | 뽕망치

Java-Spring Boot 음성 커뮤니티에서 답변·댓글·좋아요·채택과 이미지·음성 파일 업로드를 구현했습니다. S3 업로드에 UUID 파일명과 파일 유형 검사를 적용했고, 로컬 가상 데이터 10,000건으로 JMeter 부하 조건별 인덱스 적용 전후를 비교했습니다.

## 개인 프로젝트 · AI 도구 활용

| 프로젝트           | 기획·설계와 현재 구현 범위                                                                                   |
|----------------|---------------------------------------------------------------------------------------------------|
| AI 도구 맞춤 스킬 4종 | 작업별 요구사항·판단 기준, AI 도구의 범위·권한·결과물 기준을 정의했습니다. 정상·예외 상황별 테스트 시나리오와 기대 동작을 작성하고 Git으로 변경 이력을 관리했습니다. |
| 강호문파록          | 문파 경영 게임의 규칙·콘셉트·개발·테스트 방식을 정했습니다. 시드 기반 데이터 생성, 주간 진행, 전투 편성 검증과 전투 기록 재생을 포함한 프로토타입입니다.         |
| AI 게임 개발 실험    | AI를 활용한 게임 개발 방식과 개선점을 살펴보는 실험의 목적·범위를 정하고 상태 전환, 중복 입력, 내보내기 결과의 점검 항목을 정리했습니다.                  |

### AI 도구 활용 범위

게임 코드 구현·수정과 스킬 문서 초안 작성에는 AI 도구를 사용했습니다.