PJ

박재현

프론트엔드 개발자

아이디어를 실제로 쓰이는 서비스로 만듭니다

모의투자 서비스 오르자와 시세 조회 서비스 쌀스피를 기획해 출시하고 운영하고 있습니다.
오르자는 누적 이용자 **약 7,000명**과 모의거래 체결 **약 1만 건**, 쌀스피는 누적 이용자 **약 2,200명**을 기록했습니다.
금천구립 복지관 업무관리시스템은 담당자가 쓰던 엑셀 양식과 기록 방식을 바탕으로 개발해 납품했습니다.

도전 다음 적응, 단순 구현에서 끝내지 않겠습니다. ✨

서비스에 필요한 기술을 배우고 적용하며, 운영 중 발견한 문제를 개선하는 개발자입니다.
프론트엔드 개발에서 Next.js API와 데이터 수집·주문 감시까지 개발 범위를 넓혀 왔습니다.
쌀스피에서는 조회 지연의 원인을 찾아 수집과 HTTP 처리를 분리하고, 판매·구매 통계의 중복 매물 조회를 38회에서 19회로 줄였습니다.

About

2000.09.24

bliny24@gmail.com

github.com/pj6577

funfly.my/about

Skills

Frontend	Next.js · React · TypeScript · Vite · Tailwind CSS
Backend · Data	Node.js · Firestore · Supabase · SQLite · GraphQL
UI · Library	Lightweight Charts · @dnd-kit · SheetJS
Deploy · Tools	Vercel · Railway · Git

주요 문제 해결 사례

금천구립 복지관 · 서로 다른 엑셀 서식과 실적 집계 기준 2쪽

기존 기록을 보존하며 자료를 이전하고, 서비스 이용 건수와 고유 이용 인원을 구분했습니다.

오르자 · 반복 주문 조회와 순서가 뒤바뀌는 호가 수신 3-4쪽

주문 감시를 변경 구독과 메모리 탐색으로 바꾸고, 전체 호가와 실시간 변경분을 갱신 번호로 연결했습니다.

쌀스피 · 수집 중 조회 지연과 항목 전환 시 화면 초기화 5-6쪽

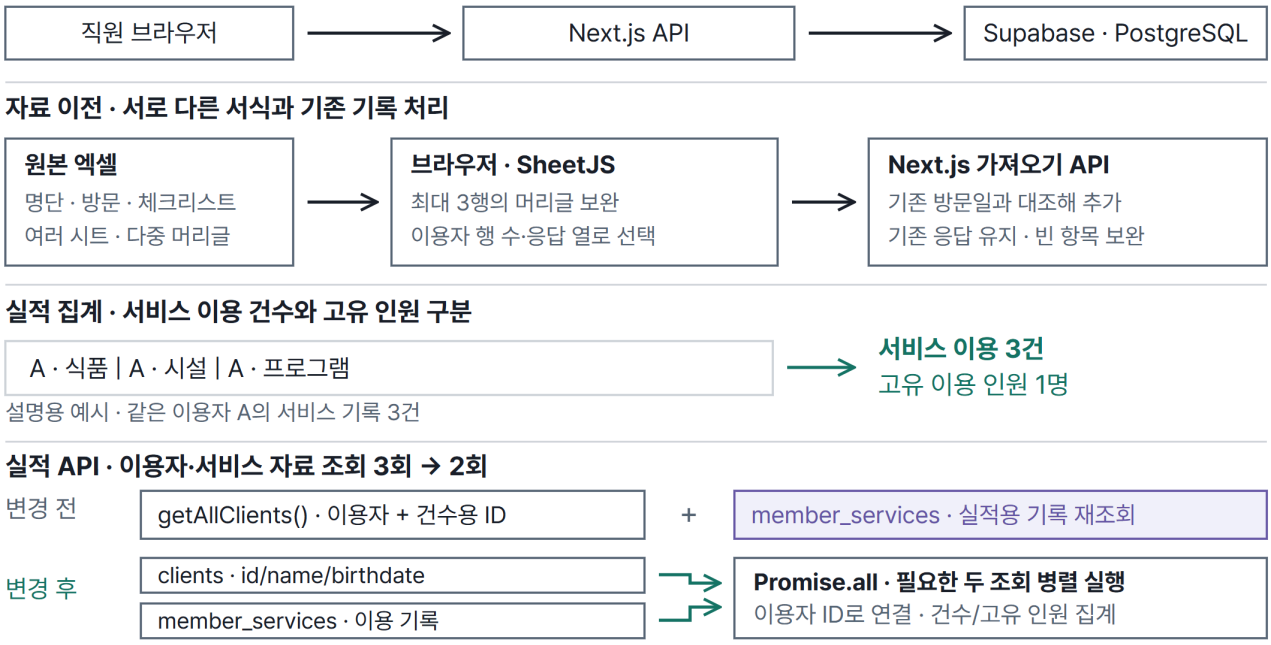
수집과 HTTP 처리를 분리하고, 대시보드를 유지한 채 조회 데이터를 교체하도록 바꿨습니다.

[서울시] 금천구립 복지관 업무관리시스템

금천누리종합사회복지관 · 2026.05 - 2026.06 · 이용자·직원 화면·API 개발·납품

이용자 기록과 월별 실적을 관리하는 복지관 직원용 업무관리시스템

엑셀 서식·집계 단위 차이를 반영한 자료 이전과 실적 조회



문제 원인

- 담당자가 사용하던 엑셀은 명단·방문·체크리스트가 여러 시트로 나뉘어 있었습니다.
  첫 행의 머리글과 이용자 행 수만으로는 열 이름을 파악하고 응답이 많은 시트를 고르기 어려웠습니다.
- 기존 실적 API는 이용자 조회 과정에서 서비스 건수를 읽은 뒤, 집계를 위해 같은 서비스 기록을 다시 조회했습니다.
  한 번 방문해 여러 서비스를 이용하므로 방문 횟수·서비스 건수·고유 인원의 집계 기준도 구분해야 했습니다.

해결 과정

- 기존 자료를 다시 입력하지 않고 이전할 수 있도록 SheetJS로 머리글을 최대 3행까지 보완했습니다.
  이용자 행 수가 같은 시트는 체크리스트 열 수를 비교해 선택했습니다.
- 자료를 다시 가져올 때 clients.access_logs의 기존 방문일은 Set으로 대조하고, 체크리스트는 기존 응답을 보존하며 빈 항목만 채웠습니다.
  브라우저에서 추출한 행은 400건씩 순차 전송하고 처리 결과를 합산했습니다.
- 실적 집계에 필요한 이용자 id·name·birthdate와 서비스 기록만 Promise.all로 읽어 Map으로 연결했습니다.
  서비스 건수는 기록의 행 수, 고유 인원은 ID Set 크기로 계산하고 입퇴실은 별도 세션으로 유지했습니다.

테스트

- 복수 시트·3행 머리글과 합성 401행의 엑셀을 사용한 함수 시험에서 열 매핑과 400+1 순차 전송을 확인했습니다.
- Supabase 응답을 모의 데이터로 대체해 같은 이용자의 여러 서비스 기록에서 월별 서비스 건수와 고유 인원이 따로 집계되는지 확인했습니다.

결과

- 기존 명단·방문 기록을 이전하고 체크리스트의 빈 응답을 보완하면서 기존 응답은 유지할 수 있습니다.
- 실적 집계에 필요한 이용자·서비스 조회를 3회에서 2회로 줄였습니다.
  직원은 개인별 이력·기간별 실적을 확인하고 업무별 8개 시트를 엑셀로 내보낼 수 있습니다.

오르자 모의투자

서비스 보기

2026.04 - 현재 · Next.js 웹·주문/계좌 API, React·Vite 앱인토스, Node.js 워커 개발·운영

누적 이용자 약 7,000명 · 모의거래 체결 약 1만 건

실시간 시세·호가를 보며 시장가·지정가 주문을 넣는 모의투자 서비스

주문 반복 조회를 변경 구독·메모리 감시로 교체



문제 원인

- 워커 도입 뒤에도 매분 Cron 3개와 기본 5분 전체 재조회가 남아 감시 대상을 반복해서 읽었습니다.
  기존 워커·Cron 설정에는 주문 변경 구독과 중복되는 감시 경로가 있었습니다.
- 시세가 지정가에 도달했더라도, 체결 처리 전에 주문이 취소되거나 다른 거래로 계좌 잔고가 달라질 수 있습니다.
  따라서 메모리에 보관한 주문 정보만으로 체결하지 않고, DB의 최신 주문 상태와 계좌를 다시 확인하도록 했습니다.

해결 과정

- 감시할 주문·포지션을 모은 인덱스의 변경을 구독해 종목별 메모리 목록을 유지했습니다.
  시세가 바뀌면 메모리에서 체결 조건을 확인하고, 같은 목록을 읽던 Cron·기본 전체 재동기화는 제거했습니다.
- 체결 조건 확인은 메모리에서, 실제 체결은 DB의 최신 상태를 기준으로 처리했습니다.
  Railway의 Node.js 워커가 Firestore 트랜잭션에서 주문의 체결 대기 상태와 계좌를 재확인하고 체결·포지션·감시 인덱스를 함께 갱신합니다.
- Next.js API도 증거금·수수료 확보와 원본 주문·감시 인덱스 저장을 같은 트랜잭션에서 처리해 감시 대상이 함께 반영되도록 했습니다.

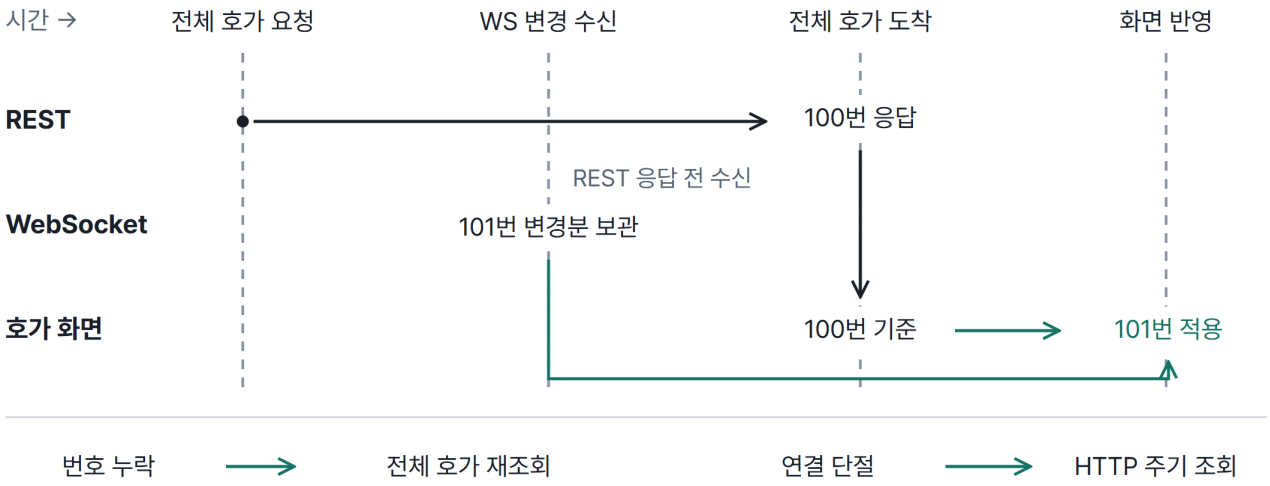
테스트

- DB 조회·구독을 모의 객체로 대체하고 실제 인덱스 모듈의 추가·수정·삭제 반영을 확인했습니다.
  초기 조회 3회 뒤 메모리에서 체결 조건을 1,000회 확인하는 동안 추가 DB 조회는 0회였습니다.
- 트랜잭션을 모의 객체로 대체해 executor에 취소·기체결·존재하지 않는 주문을 전달했습니다.
  체결 데이터를 쓰지 않고 감시 인덱스를 정리하는지 확인했습니다.

결과

- 매분 Cron 3개·기본 5분 전체 재조회 경로를 없애고, 사용자가 접속을 종료한 뒤에도 워커가 지정가 주문 감시를 이어갑니다.
- 메모리에서 체결 조건을 충족해도, DB의 최신 주문 상태와 계좌를 확인한 뒤 거래를 반영합니다.

도착 순서가 다른 호가 응답을 갱신 번호로 동기화



문제 원인

- 앱인토스에서 REST 전체 호가를 기다리는 동안 WebSocket 변경분이 먼저 도착할 수 있습니다.
  전체 응답과 변경분을 도착 순서로 덮어쓰면 최신 수량이나 중간 변경을 놓칠 수 있었습니다.
- 체결 알림은 계좌 갱신 신호이므로 알림만으로 잔고를 계산하지 않고 확정된 계좌를 다시 읽어야 했습니다.

해결 과정

- 초기 WebSocket 변경분을 버퍼에 보관하고, 전체 호가의 lastUpdateId 이후 변경분만 순서대로 적용했습니다.
  수량이 0이 된 가격대는 호가 목록에서 삭제했습니다.
- 번호가 빠지면 변경 이력을 이어 붙일 수 없어 전체 호가를 다시 읽도록 했습니다.
  WebSocket 연결이 끊기면 5초 간격의 HTTP 조회로 전환하고 2.5초 뒤 재연결을 시도합니다.
  재연결되면 HTTP 주기 조회를 멈추고 전체 호가부터 다시 동기화합니다.
- 계좌는 체결 알림·재접속·화면 복귀 시 재조회하고 연속 알림은 묶었습니다.
  변경 시각·활성 목록 버전이 같으면 주문·포지션·거래 목록의 재조회는 생략합니다.

테스트

- WebSocket·HTTP 응답을 모의 데이터로 대체해 101번 변경 뒤 100번 전체 호가가 도착해도 최신 수량이 유지되는지 확인했습니다.
  102→105 번호 누락의 재조회와 연결 복구도 확인했습니다.
- 앱 계좌 구독 함수의 연속 알림 3개가 700ms 뒤 조회 1회로 묶이는지 확인했습니다.
  숨김 상태는 조회를 생략하고 화면 복귀 시 다시 실행했습니다.

결과

- 전체 호가 이후의 변경분을 순서대로 반영하고, 번호 누락·연결 단절은 전체 재조회와 대체 조회로 복구합니다.
- 접속하지 않은 동안의 거래도 재접속 시 계좌에 반영해 보여주며, 변경 없는 하위 목록은 다시 읽지 않습니다.

쌀스피 · 시세 조회 서비스

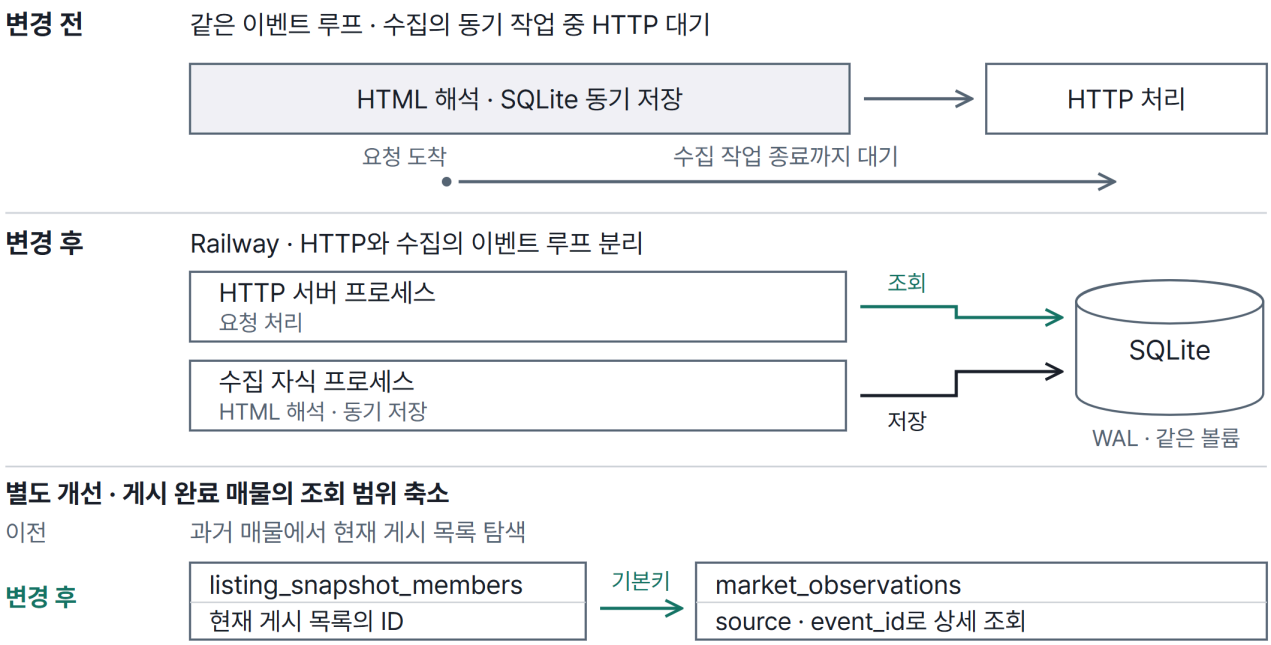
서비스 보기

2026.10 - 현재 · Next.js 시세 조회 웹·수집·조회 API 개발·운영

누적 이용자 약 2,200명 · 완료·종료 수집 기록 약 29만 건

여러 거래소의 디지털 재화 시세·매물·거래 내역을 조회하는 서비스

시세 조회 지연의 원인을 추적해 수집과 조회를 분리



문제 원인

- 운영 요약의 캐시 미적중(CDN MISS) 응답 11,025ms를 관측했습니다.
  별도 프로세스의 DB 읽기만으로 지연이 설명되지 않아 HTTP 경로를 확인했고, HTML 해석·동기 SQLite 저장이 API와 같은 이벤트 루프에서 실행되고 있었습니다.
- 현재 매물도 과거 누적 기록에서 찾고 일부 행마다 최신 시각을 확인했습니다.
  판매·구매 통계가 같은 매물을 따로 읽어 19개 항목의 매물 조회가 38회 발생했습니다.

해결 과정

- Vercel의 웹·공개 API와 Railway의 조회 서버를 유지하고 수집을 자식 프로세스로 분리했습니다.
  파싱·저장이 HTTP와 다른 프로세스에서 실행되도록 하되 같은 볼륨의 SQLite WAL DB를 공유했습니다.
- 현재 매물 조회는 listing_snapshot_members에서 시작해 source·event_id 기본키로 상세를 읽도록 바꿨습니다.
  필요한 현재 ID 목록을 먼저 정해 과거 누적 기록을 탐색하던 조회 순서를 바꿨습니다.
- 판매·구매는 같은 매물 목록에서 계산할 수 있어 목록을 한 번 읽고 두 통계를 함께 구했습니다.

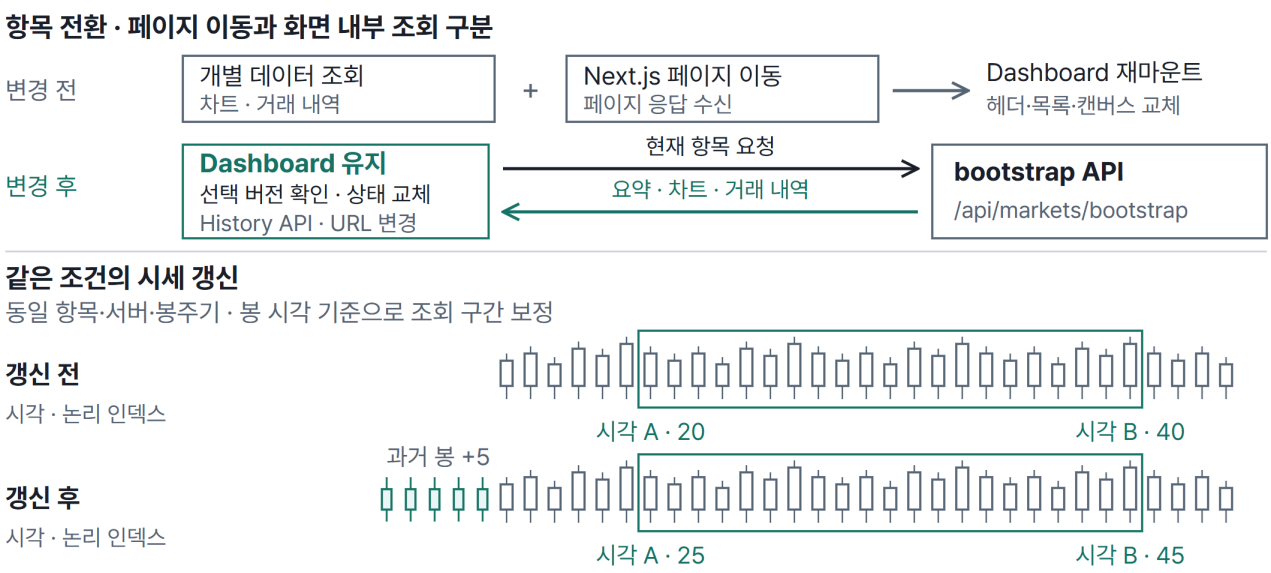
테스트

- 로컬 Node.js·임시 DB에서 수집 자식의 3초 CPU 작업 중 /health가 1.5초 제한 내 응답하는지 확인했습니다.
- 과거 매물 20,000건의 임시 SQLite에서 전체 SCAN 대신 기본키 SEARCH를 확인했습니다.
  기존 DB의 19개 항목·57개 목록 비교 기록에서도 변경 전후 결과가 같았습니다.

결과

- HTTP 처리와 수집의 HTML 해석·동기 저장을 별도 프로세스에서 실행하도록 나눴습니다.
- 19개 항목의 판매·구매 요약에 필요한 매물 조회 호출을 38회→19회로 줄이고, 현재 수집 목록의 ID로 상세를 찾도록 변경했습니다.

항목 전환·시세 갱신 때 화면 재생성을 줄이고 조회 구간 유지



문제 원인

- 운영 화면 전환을 추적하니 개별 차트·거래 내역 요청과 Next.js 페이지 이동이 겹쳐 헤더·대시보드·차트 캔버스가 교체됐습니다.
- 새 시세에도 차트가 재생성돼 확대 구간이 초기화됐습니다.
  과거 봉이 앞에 추가되면 같은 위치 번호가 다른 봉을 가리켜, 위치 번호만 복원해도 보던 구간이 달라졌습니다.

해결 과정

- 목록·차트·거래 내역을 한 번에 받는 bootstrap API로, 항목 전환 시 대시보드를 유지하고 데이터만 교체했습니다.
  URL은 History API로 바꾸며 직접 진입·새로고침은 SSR과 메타데이터를 유지했습니다.
- 이전 요청을 취소하고 현재 선택에 해당하는 응답만 반영해, 항목을 연속 선택해도 이전 응답이 화면을 덮어쓰지 않도록 했습니다.
  조회 중이거나 실패했을 때 새 항목 이름 아래 이전 가격을 표시하지 않습니다.
- 같은 조건의 새 시세는 보던 구간을 유지하고 항목·서버·봉주기가 바뀔 때는 최근 구간을 보여주도록 기준을 나눴습니다.
  Lightweight Charts의 데이터는 setData()로 교체하고 기존 봉의 시각으로 위치를 보정하며, 종류·테마 변경에만 차트를 재생성합니다.

테스트

- 브라우저에서 연속 선택·지연 응답·실패 후 재시도·뒤로/앞으로를 재현했습니다.
  현재 선택의 응답만 반영되고 기존 DOM·캔버스가 유지되는지 확인했습니다.
- 합성 봉 100개를 사용한 갱신 effect 시험에서 앞에 5개를 추가해, 같은 시각의 구간을 가리키는 논리 인덱스가 20~40에서 25~45로 보정되는지 확인했습니다.

결과

- 항목을 전환해도 헤더·대시보드·차트 인스턴스를 유지하며, URL 직접 접속은 SSR로 처리합니다.
- 새 시세와 과거 봉이 추가돼도 보던 시각의 확대 구간을 유지하고, 조회 조건이 달라지면 최근 구간·가격 축을 다시 맞춥니다.

오늘시작 · 채용 지원 기록 관리 서비스 보기

React·Dexie·IndexedDB 기반 지원 기록·메모·일정 관리 Chrome 확장 프로그램
설치 수 200명 이상 · GitHub Insights Traffic Clone 350건 이상