FastAPI 특강

5일차: FastAPI 미니 프로젝트 ✨

목 차

- 4일차 복습
- FastAPI 프로젝트

4일차 복습

WebSocket이 필요한 이유

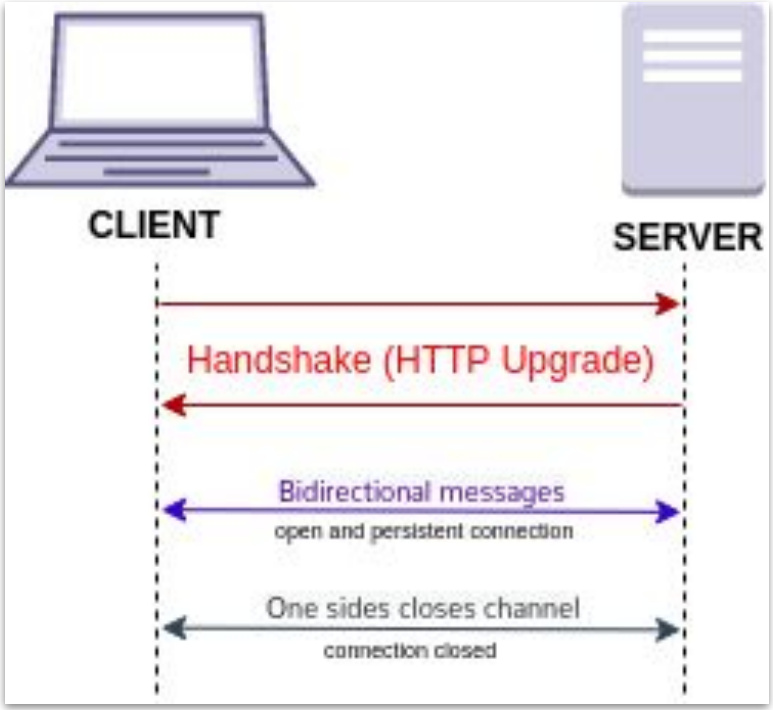
= 기존 HTTP 방식의 문제점

- 실시간성 부족
  - HTTP 요청-응답 방식은 클라이언트가 요청해야만 서버가 응답 가능
  - 실시간 알림, 채팅 등에는 비효율적
- 풀링(Polling)과 긴 연결(Long Polling)의 한계
  - 풀링: 클라이언트가 일정 시간마다 서버에 요청 (과부하 문제)
  - 긴 연결: 클라이언트가 연결을 유지하며 서버 응답을 기다림 (비효율적)

HTTP vs WebSocket 비교

항목	HTTP	WebSocket
연결	요청마다 새로	한 번 연결 유지
통신	단방향 (Client -> Server)	양방향
사용 예시	게시판, 블로그	채팅, 알림, 실시간 주식

WebSocket 프로토콜 구조



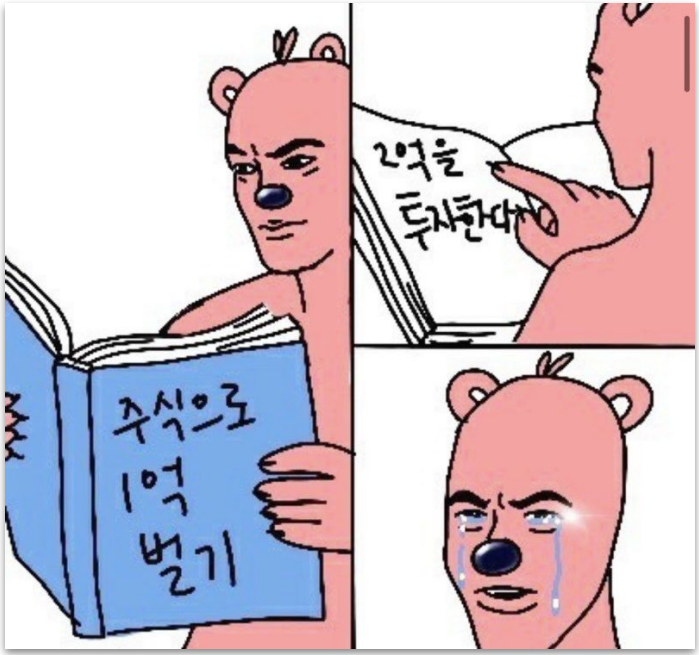
프로젝트

🎯 목표

- FastAPI로 실시간 모의 투자 시스템의 핵심 CRUD 및 비즈니스 로직 구현
- SQLite + SQLAlchemy를 활용한 유저 자산 및 포트폴리오 DB 연동
- WebSocket을 이용한 실시간 시세 스트리밍 이해 및 구현
- JWT(OAuth2) 기반의 보안 인증을 통해 나만의 자산 데이터 보호하기

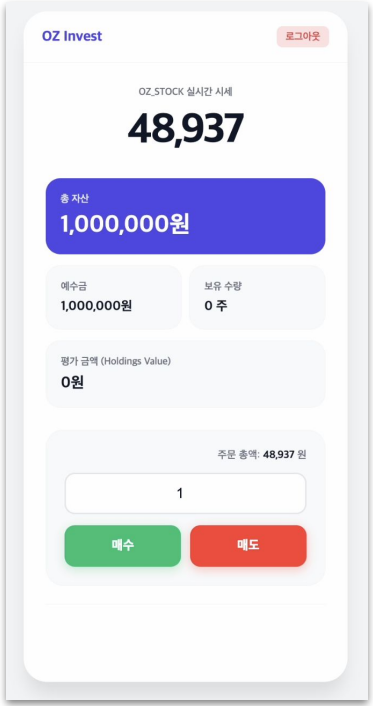
📦 프로젝트 개요

- 주제: 실시간 모의 투자 시뮬레이터



📦 프로젝트 개요: 실시간 모의 투자 시뮬레이터

- 데모



https://youtu.be/J0wlWj2jLZw?si=dqUj2VNipg4W9bB5

📦 프로젝트 개요: 실시간 모의 투자 시뮬레이터

- **주요 로직**
  - 인증(Auth): Argon2 해싱을 이용한 회원가입 및 JWT 토큰 기반 로그인
  - 매수 (Buy)
    - 보유 현금 >= 주문 총액
    - 잔액 부족 예외 처리
    - 평균 매수 단가(Avg Price) 실시간 계산 및 업데이트

📦 프로젝트 개요: 실시간 모의 투자 시뮬레이터

- **주요 로직**
  - 매도 (Sell)
    - 보유 수량 >= 매도 수량
    - 수량 부족 예외 처리
    - 매도 후 잔액 입금 및 포트폴리오 데이터 최적화

📦 프로젝트 개요: 실시간 모의 투자 시뮬레이터

- **주요 로직**
  - 실시간 자산 계산 (Read)
    - 평가 금액 = 보유 수량 x 현재가
    - 평가 손익 = 평가 금액 - (보유 수량 x 평균 단가)
    - 총 자산 = 현금 + 평가 금액

📦 프로젝트 개요: 실시간 모의 투자 시뮬레이터

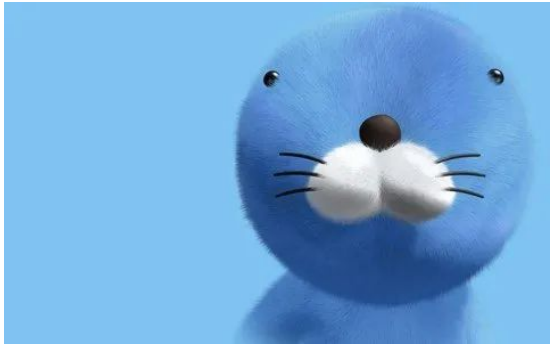
- **최종 API 구조**
  - POST /register: 회원가입 (비밀번호 암호화)
  - POST /login: 로그인 (JWT 토큰 발급)
  - GET /user/status: 내 실시간 자산 및 손익 현황 조회
  - POST /trade/buy: 주식 매수
  - POST /trade/sell: 주식 매도
  - WS /ws/market: 실시간 시세 브로드캐스팅

📦 프로젝트 개요: 실시간 모의 투자 시뮬레이터

- **추천 진행 순서**
  - FastAPI 와 SQLite 연동 후 User 및 Portfolio 테이블 설계
  - asyncio를 활용해 랜덤하게 가격을 생성하고 WebSocket 으로 전송
  - 매수/매도 시 발생할 수 있는 예외 상황(ex. 돈 부족)을 완벽하게 처리

📢 발표자 리스트

- [2시 30분] 첫 번째 발표자: ??
- [2시 40분] 두 번째 발표자: ???
- [2시 50분] 세 번째 발표자: ???



<발표자 한정 공지>

- 작성하신 코드를 화면공유를 통해 보여주세요. 미완성본이라도 괜찮습니다.
  왜 이렇게 코드를 구현했는지를 중심으로 진행해보세요!!
- 서버 실행 후 실제 프로젝트가 돌아가는 과정을 공유해주시면 더욱 좋습니다.

QnA