

# 수행 프로젝트 상세 설명

이름 : 정순구

이메일 : [soon.good@outlook.com](mailto:soon.good@outlook.com)

## 웨이커 (2021.06 ~ 2022.06)

- 신규 서비스 API 개발
- 미국주식 실시간 데이터 처리 애플리케이션 개발

### 1.신규서비스 API 개발

기존 레거시 서비스를 개선하여 신규서비스의 API를 개발했습니다. 테이블을 설계하고, API 설계/구현하는 업무가 주된 업무였습니다.

대가들의 분석, 뉴스, 주요 재무, AI 종목 분석, 종목정보 등 주식 탭에 존재하는 API를 개발하는 업무입니다.

참고)

<https://waiker.ai/stock?ric=MSFT.O>

### 2. 미국주식 실시간 데이터 처리 애플리케이션 개발

미국주식은 주식 거래 데이터를 전송해주는 데이터 벤더사가 있습니다. 이 벤더사 측에 TCP/IP로 통신을 하고 데이터가 올때마다 해당 데이터의 일/시/분/초 데이터를 각 기간별 고가/저가/시가/종가를 계산해서 저장하는 것과 DB 인서트 작업을 처리하는 서비스였습니다. 트래픽은 초당 5000건 ~ 8000건 정도였습니다.

기존 데이터 애플리케이션에서의 단점을 파악해 신규 서비스 개발시 단점을 보완하기 위해 생산자/소비자 모델을 도입해 이 단점을 해결했고, 프로젝트를 75% 정도 마무리지었습니다.

프로젝트를 신규로 진행하면서 결과적으로는 아래의 목표들을 달성했습니다.

### **트래픽 처리 구조 개선**

- 병렬구조 개선, 생산자/소비자 모델 설계/구현
- 나스닥 1만 2000건 수신 처리
- RabbitMQ 도입

### **레거시 프로젝트 틱 데이터 처리 오류 발견/검증 및 수정/리팩토링**

- 단위테스트 기반 레거시 코드 검증
- TDD 기반 틱 데이터 처리 코드 개발/리팩토링/검증

### **테스트 커버리지 개선, 개발 Phase 인프라 필요성 어필, 인프라 구성/셋업**

- 계층별 테스트 패키지 레이어 구성/설계
- 개발 Phase 인프라 필요성 어필, 인프라 구성/셋업

### **기억에 남았던 업무**

#### **TDD 기반 틱 데이터 처리 코드 개선**

외주사 개발로 운영되었던 레거시 코드인 valuesight 서비스의 틱데이터를 처리하는 로직이 버그가 있음을 인지하고, 신규 프로젝트 개발시, 테스트 코드 기반으로 검증해서 새로운 틱 계산 로직을 구현했습니다. 이 후 프로젝트 전반적으로 산발되어있던 단위테스트를 패키지 레이어 단위로 재구성하고 설계하는 과정을 겪었습니다. 이 과정에서 테스트를 구성하면서 테스트의 경계를 명확히 해야 하는 것의 필요성을 느꼈고, 한 테스트에서 필요한 가짜객체의 역할 등을 새롭게 다시 깨닫게 된 계기였습니다.

#### **나스닥 1만 2000건 트래픽 처리/병렬구조 개선**

나스닥 1만 2000건 처리를 시작하면서, 중국주식 트래픽과는 다르게 트래픽의 빈도가 매우 잦았고 데이터 저장속도와 웹소켓 송/수신 속도가 트래픽의 속도/빈도를 따라가지 못하는 이슈가 있었습니다. 중국주식 트래픽을 처리할 때 비동기 풀을 사용하던 기존의 단순한 방식으로는 데이터 저장이 쉽게 이뤄지지 않는 맹점이 있었습니다.

주어진 시간 안에 이 문제를 해결하기 위해 기존과는 다른 방식의 접근이 필요했습니다..

여러가지의 요소들을 고려해본 결과, 인프라 설계, 생산자/소비자 설계, 로직 변경을 했고, 이에 따른 리팩토링 작업을 수행했습니다. MSA 전환 가능성도 염두에 두고 작업의 성격별로 아래의 3가지의 모듈로 분리했고, 각각을 다른 인스턴스에서 동작하도록 안정성을 확보했습니다.

- data-live 모듈: TCP 데이터수신처리 모듈,
- data-collector 모듈: Tick 데이터 계산, 웹소켓처리, 실시간 DB INSERT/UPDATE 모듈
- data-websocket 모듈 : 웹소켓 MQ 데이터 수신/웹소켓 송신 모듈

분리 작업 이후 트래픽 처리를 위해 data-collector 모듈에 대해 기존과는 다르게 트래픽을 처리할 수 있도록 생산자/소비자 모델을 설계하고 직접 구현했습니다. 생산자 측에서는 MQ에서 수신한 데이터를 작업으로 변환 후 캐시에 쌓아두고, 소비자 측에서는 일정 주기에 한 번씩 특정 사이즈 만큼의 작업을 꺼내서 데이터를 저장/MQ 푸시를 하는 구조였습니다.

이때 작업을 생산한 것을 제대로 소비자 측에서 소비하는 등의 작업 들을 TDD 기반으로 구현했고, 테스트 커버리징 단계에서 단위테스트를 계층테스트로 전환하면서 테스트가 용이한 코드의 구조에 대해 다시 깨닫게 된 계기였습니다.

## 디케이테크인 (2019.11 ~ 2021.06)

상용서비스를 운영/유지보수/리팩토링하는 업무였습니다. 주로 멜론 음원서비스 파트 내의 댓글/HiFi 를 주로 담당해서 유지보수/개발 업무를 담당했고 때에 따라서 다른 분들의 업무를 지원하는 업무를 맡았습니다. 경험했던 주요 내용들은 아래와 같습니다.

### [뮤직 DNA 6.0 개편]

- 월간/주간 개인 선호음악/추천음악 조회 백엔드 API 개발

### [피드, 마이로그 6.0 개편]

- 개인화 영역 소식 내역 메시지 생성 기능 개발

#### [멜론 댓글 서비스 운영/유지보수]

- 슬로우 쿼리 유지 보수/대응
- 카카오 클린 플랫폼 연동 블랙리스트 기능 개발

#### [멜론 HiFi/음원/스테이션/댓글/카카오뮤직/계정 운영/유지보수]

- 멜론 스테이션 배너 앱 랜딩 이슈
- Sandbox MongoDB 샤드 클러스터링 전환 작업
- 웹뷰 버그 수정/이슈 대응, 다크모드 지원
- 카카오 계정 통합 관련 이슈 지원

#### 기억에 남았던 업무

멜론 댓글 서비스 개발/운영시에 했던 업무들이 가장 기억에 남습니다. 슬로우 쿼리 문제를 해결하면서 직접 상용 DB의 구조를 파악해야 했는데, 테이블 조인 관계를 줄이는 과정에서 제품에 관련된 테이블의 설계를 치밀하게 파악해가는 과정에서 묘한 흥미와 도전의식을 느꼈습니다. 이 외에도 블랙리스트 기능 개발 등 여러가지 재미있는 개발 업무들이 주어져서 도전적으로 업무를 수행할 수 있었던 좋은 기회였다고 생각합니다.

## 누리플렉스 (2018.10 ~ 2019.11)

SK E&S, SUSTERA 등의 사업장의 에너지 모니터링 솔루션 개발/운영업무를 담당했습니다. 담당했던 주요 업무들은 아래와 같습니다.

#### [SK E&S STEP 에너지 모니터링 솔루션 개발/운영업무]

백엔드 개발

- EMS/TOC API 개발
- 장비 개별 정보 현황/이력 데이터 처리 로직 설계/구현

#### 프론트엔드 개발

- 차트 라이브러리 연동 데이터 시각화
- SockJS 연동 웹소켓 데이터 처리 로직 개발

#### [SUSTERA PMS 에너지 모니터링 솔루션 개발/운영]

##### 백엔드 개발

- PMS 세부 API 개발
- 장비 개별 정보 현황/이력 데이터 처리 로직 설계/구현

##### 프론트엔드 개발

- 차트 라이브러리 연동 데이터 시각화

#### [CJ / 부산 신재생 에너지 혁신센터 EMS 운영/유지보수]

- 실시간 데이터 정합성 에러 파악, 트러블 슈팅

#### 기억에 남았던 업무

개인적으로 가장 기억에 남았던 것은 장비 개별정보 현황/이력 데이터 처리 로직을 개발하는 것이었습니다. BMS 나 RACK 등의 장비들의 구조들을 파악한 후에 SQL을 작성해야 했고, JSP/JSTL에 데이터를 바인딩하기 위해 데이터 구조 역시도 만들어야 했는데 실제 장비의 특성을 파악하면서 개발을 해야 했기에 어려우면서도 가장 기억에 남았던 기억이었습니다.

## 주니코리아 (2015.05 ~ 2017.04)

3G/4G 무선 신호를 Wifi로 변환해주는 기업용 데이터 라우터들을 모니터링하고 관리/제어하는 솔루션을 개발했습니다. 담당했던 주요 업무들은 아래와 같습니다.

Hardware Replacement

= 장비 설정 파일(xml) SFTP Upload, Download 기능 개발

= GUI (프로그래스 바) 컴포넌트 데이터(다운로드 진행율)비동기 처리 로직 개발

HEMS 유지보수/개발/운영

QA 대응, 유지보수/개발/운영, 트러블슈팅

## **기억에 남았던 업무**

GUI 컴포넌트 데이터 처리, SFTP Upload/Download 기능을 개발하면서 프로그래스 바 UI를 렌더링 하면서 동시에 파일이 몇 퍼센트 정도 다운로드 되었는지, 업로드 되었는지를 표현하는 기능을 개발하는 등 입사초기에 해볼 수 있는 재미있는 경험을 할 수 있었기에 좋은 경험이었다고 생각합니다.