



이준영 Junyoung Lee

☎ 010-7210-2992

✉ 8byte@naver.com

🐙 <https://github.com/64byte>

📄 <https://www.acmicpc.net/user/8byte>

🏠 <https://leetcode.com/8byte/>

간략한 소개

대용량 트래픽 처리 및 다양한 서비스를 경험하고 싶은 신입 개발자입니다.

중학교 시절 온라인 게임 서버를 시작으로 프로그래밍에 흥미를 갖게 되었고, 웹을

통해 다양한 서비스를 실현할 수 있다는 것을 느끼고 난 지금은 웹서버 개발자를

꿈꾸고 있으며, 웹 동작 방식이나 사용 기술에 대해 관심이 많습니다.

프로그램은 혼자 만들 수 없다는 걸 느끼며 개발하고 있습니다.



보유기술

(관심 분야)

Programming Languages (프로그래밍 언어)

- C, C++11, C# 7.0, Python3.8, Javascript, Java, HTML5, CSS3

Database (데이터베이스)

- Maria DB(MySQL), Oracle

Environment (환경)

- AWS Lightsail, Windows, Linux(Ubuntu, Rasbian)

Current Interest Fields (현재 관심 분야)

- Django, ExpressJS, Flask, Electron, React, GraphQL

Previous Interest Fields (이전 관심 분야)

- Go, Rust, Mongo DB, WPF, PHP, Xamarin, Android

프로젝트

IoT 플랫폼 개발 프로젝트(미세먼지 감시 시스템) - 2018.01 ~ 2018.02

- 개요: 사용자가 미세먼지를 측정하고 스마트폰과 연동하여 측정 데이터를 웹서버에 저장하고 데이터는 안드로이드 애플리케이션 화면과 웹 화면을 통해 조회
- 팀 구성: 웹 개발 (2), 안드로이드 개발 (2), 임베디드 개발 (1)
- 역할: 웹 개발
- 사용 스택: PHP, Slim framework, JWT, Javascript, D3.js, Maria DB
- 업무
 - 요구 사항 정의
(<https://trello.com/b/H4h41yy8/%EC%9A%94%EA%B5%AC-%EC%82%AC%ED%95%AD>)
 - 기능 정의 (<https://trello.com/b/4OEbonfx/function>)
 - 설계
 - 구현
 - I. JWT 를 이용한 회원가입 이메일 OTP 인증 구현
 - II. CSRF 토큰을 이용한 요청 위조 방지 구현
 - III. D3.js 를 이용한 실시간 미세먼지 변화 그래프 조회 구현
 - IV. Restful API 구현
 - V. JWT 를 이용한 API 호출 사용자 검증
- 저장소 주소: <https://github.com/64byte/teamb-iot>

시각장애인을 위한 미세먼지 알리미 - 2018.04 ~ 2018.05

- 개요: 시각장애인으로 실내 미세먼지를 알 수 있도록 Clova TTS API 를 통해 정보 제공
 - 팀 구성: 임베디드 개발 1, 안드로이드 개발 1, 디자인 1
-

- 역할: 임베디드 개발, 안드로이드 개발
- 사용 스택: Python, Java, Android
- 업무
 - 요구 사항 정의
 - 기능 정의
 - 구현
 - I. 라즈베리파이와 연결된 미세먼지 센서 및 온/습도 센서의 동작 구현
 - II. 라즈베리파이와 안드로이드 블루투스 연결 구현
 - III. Clova TTS Rest API 호출을 이용한 시각장애인용 안드로이드 어플리케이션 구현
 - IV. 일반 사용자용 어플리케이션 구현

문제해결/개선 사례

1. 여러 태스크가 하나의 컨테이너에서 데이터를 꺼낼 때 중복 사용하는 문제 - 2019. 05

- 문제: C#의 Task 자료구조를 이용하여 다수의 태스크를 생성하여 하나의 컬렉션에서 데이터를 중복으로 사용하는 문제
- 원인: Task 자료구조가 단일 스레드에서 동작하는 줄 알았으나, Task 자료구조 명세서에 따르면 유기적으로 다중 스레드에서 동작하는 방식
- 대안: 스레드로부터 안전한 컬렉션 구현
- 해결
 - 1) lock 을 이용한 스레드 동기화
 - 2) Interlocked 으로 구현된 ConcurrentCollection 사용
- 평가: 처음 lock 을 이용한 스레드 동기화로 문제를 해결했지만, 태스크가 많아질수록 잦은 busy-waiting 으로 인한 성능 이슈로 인하여 Interlocked 으로 구현된 ConcurrentCollection 을 사용하여 성능을 향상함

2. 짧은 시간에 많은 API 웹서버 HTTP 요청에 대한 지연 개선 - 2020. 02

- 개요: 데이터를 저장하고 꺼내는 간단한 웹 API 서버, 데이터가 저장된 시점으로부터 특정 시간이 지나면 지워지도록 구현
- 문제: 웹 API 서버가 초당 200+번의 요청을 처리하지 못해 지연이 생김
- 원인: 처음 웹 API 서버는 Python 의 Flask 로 구현, 그러나 해당 프레임워크가 초당 많은 요청을 처리하지 못함 (<https://github.com/the-benchmark/web-frameworks> / 위 내용은 해당 페이지를 참조함)
- 대안: 서버 증설, 구조 변경 혹은 프레임워크/언어 변경
- 해결
 - 1) Nodejs 의 Express 로 API 서버를 다시 작성

-
- 평가: 작성한 Python 코드가 다른 언어로 변환하기에 짧았고 데이터의 유효 시간을 확인하여 지우는 과정에서 스레드 동기화로 인한 지연을 생각하지 않아도 되는 Nodejs 로 변환하는 것을 결정, 코드를 바꾼 이후로는 API 요청이 초당 300+번의 요청에도 지연없이 처리되어 문제가 개선됨을 확인함

학력

동의대학교 컴퓨터공학과 학사 졸업 - 2015.03 ~ 2019.02

- 학점 4.44/4.50
- 단과대학 수석 졸업

건국고등학교 졸업 - 2008.02 ~ 2011.03

언어

영어 (Intermediate)

한국어 (Native)

기타 인적사항

OPIC (Intermediate High) - 2019.01.21 취득

정보처리기사 - 2018.09.17 취득

병장 만기 제대 - 2011.11.28 ~ 2013.08.27

여기까지 제 이력서를 봐주셔서 감사합니다.

더 궁금하신 내용이 있으시면

8byte@naver.com / 010-7210-2992 으로 연락주시면 감사하겠습니다.