

Portfolio

정리에 신경쓰려고 노력하는
백엔드 엔지니어, 유태규



유태규

생년월일

1995.06.07

학교 / 전공

한양대학교
수학교육학 전공

희망 근무지
서울/경기

정리에 신경쓰려고 노력하는 백엔드 엔지니어

프로그래밍 역량

Javascript	중상	작동원리를 이해하고 이에 따른 코드 작성 가능 기본적인 서버 구축 경험(express)
Typescript	하	기본적인 개념에 대한 이해
DB	중하	기본적인 테이블 설계 경험 검색을 통해 기본적인 sql 작성 가능 Nosql(redis) 기본적인 사용 경험 및 구축 경험
Python	중상	작동 원리를 이해하고 이에 따른 코드 작성 가능 기본적인 서버 개발 및 운영 경험(flask)
Docker	중	기본적인 사용 경험 및 원리 숙지 커스텀 이미지 생성 및 배포 경험
Linux	중	R기본적인 사용 경험(centos)(온프레미스 환경) 기본적인 쉘 스크립트 작성 경험

경력

다날 핀테크

개발본부 / 사원

2022.02 - 현재

- 회사 홈페이지(www.payprotocol.io) 관리를 포함한 정적 소스 관리
- 통합 계정 관리 솔루션(keycloak, casbin) 검토 및 데모 개발
- 웹서버 apache에서 nginx로 교체
- 자사 앱 리뷰 크롤링 및 공유용 구글 스프레드 시트 개발
- ISMS 보안 조치(docker, apache, redis 등)
- 앱에 쓰이는 웹뷰 서버 운영 및 유지보수(로그 정책 변경, 서버 의존성 제거 등)
- 회사 내부 백오피스 개발 및 유지보수(버그 픽스, 신규 관리 기능 추가 등)

#vue #nginx #html #docker #redis #flask

링크 | <https://www.payprotocol.io>

링크 | <https://app.payprotocol.io>

깃허브 및 블로그

깃허브 | <https://github.com/OTKRyu>

블로그 | <https://otkryu.github.io/>

01 이용자 보호센터

프로젝트 기간 : 2022.11 - 2022. 12

프로젝트 현황 : <https://carecenter.payprotocol.io> 에서 확인가능

이용자 보호센터

기 간 : 2022.11 - 2022.12

개 요 : 코인 이용자들이 보호받을 수 있도록, 투명하게 정보 공개할 수 있는 창구

코인 시장이 활성화됨과 동시에, 신뢰할 수 있는 코인의 기준 또한 높아졌으므로 이를 만족하기 위해서 투명한 정보 공개가 중요하게 되었다.

이를 위해서 공시 정보, 유통 계획안 및 회사에서 가지고 있는 지갑과 그 잔량 등을 공개하는 사이트를 만들 필요성이 대두되었다.

담 당 역 할	풀스택 엔지니어
기 획 배 경	코인 이용자들이 보호받을 수 있도록, 투명하게 정보 공개할 수 있는 창구
개 발 환 경	Node.js, Centos, Javascript, React, Mysql
구 현 사 항	프론트엔드 : router, ajax 통신을 통한 게시판, 시세 및 소유 코인 현황 구현 백엔드 : rest 규격에 맞는 api 포인트 구현 및 DB 통신을 통한 데이터 전달 배포 : 배포 용이하도록 container화, 서버 의존성 최소화(.env), shell script를 통한 container 옵션 문서화 사용자 인터페이스 : 백오피스에서 공개 정보 조정할 수 있도록 백오피스 기능 개발

어려웠던 점	<ol style="list-style-type: none">1. 요구사항이 대충 만들어져서 왔기 때문에, 어떤 식으로 개발하는 것을 정하는 데에 고민이 있었다. 이전의 프로젝트에서 구현한 방식을 따라했지만, 하면서도 이렇게 하는게 잘하는 짓인가하는 의문이 사라지지 않았다.2. Express 서버 개발해보는 것이 처음이었는데, 로직을 구현하는 것은 어렵지 않았지만 개발방법론이나 js 맞는 개발방식 혹은 배포방식등을 고민하고 결정하는 것이 어려웠다.
느낀 점 및 개선점	<ol style="list-style-type: none">1. 이전까지는 개발되어 있는 프로젝트를 유지보수하는 일을 주로 했었기 때문에, 신규로 개발하면서 내가 취업하기 전 학생으로서 프로젝트했을 때와 얼마나 달라졌는지 한번 점검해볼 수 있는 기회가 되었다. 인프라나 틀에는 익숙해졌다는 느낌이 확실히 들었지만, 여전히 잘한 개발이 어떤 것인지에 대해서는 명확히 알지 못한다는 생각이 든다.2. 서버 규모가 워낙 작았고, 개발 기간을 타이트하게 잡을 수 밖에 없는 환경이었기 때문에, 확장 용이성이나 테스트 등에 대해서는 신경을 많이 쓰지 못했는데, 이 부분을 고려하면서 했더라면 더 좋은 프로젝트가 되었을 거라고 생각한다.
총평	개발 기간, 개발 규모 등 작은 프로젝트였고, 확장에 대해서도 불확실한 프로젝트였다. 내 나름으로는 작은 프로젝트에 맞는 개발 방식을 취했다고 생각하지만 이게 적절한 판단인지에 대해서는 의문이 든다.