

정석진

Web Backend Developer



Career 신입

Phone 010-8936-3879

Birth 1996.10.17

Email strong3879@gmail.com

<https://github.com/strong1133>

About Me <https://www.notion.so/fa0eb946388d40b592959e29d1eca65f>

핵심 역량

- Spring Boot 기반 RestAPI 서버 개발 경험
- Python Flask 기반 RestAPI 서버 개발 경험
- React Axios 를 통한 API 요청 개발 경험
- CSS, Styled-Component 를 이용한 CSS 개발 경험
- AWS의 EC2, RDS를 활용한 서버 구축 및 배포 경험
- S3, IAM를 통한 이미지 처리 경험
- Route53, ACM, Load Balancer를 통한 SSL인증서 발급 및 HTTPS 배포 경험

경력

2019.04 ~ 2020.10 케이텍 맨파워 한양대학교 구리병원 응급 원무과

학력 및 교육

2015.03 ~ 2019.02 삼육보건대학교 의료정보과 졸업

2021.01 ~ 2021.03 스파르타코딩클럽 웹개발 종합반 수료

2021.03 ~ 2021.06 항해99 1기 수료

자격증

2019.04 ~ 기한 없음 MOS2016 Master MicroSoft

2020.12 ~ 기한 없음 컴퓨터활용능력1급 대한상공회의소

활동

2021.03 ~ 2021.05 스파르타 코딩클럽 Spring 기본반, 미니튜터 1기 활동

Level 1~5

Language Java (3), HTML(4), CSS(4), JS(3), Python(3)

Framework + Library Spring (4), React(2), Flask(3)

DataBase MySQL(3), MongoDB(3)

ETC Git(3), Slack(3), Figma(2), PostMan(3)

프로젝트

In progress

스터디, 사이드 프로젝트 팀빌딩 플랫폼, Demoim Backend 개발

향해 99 마지막 실전 프로젝트인 Demoim 의 팀 리더이자, Backend 개발, 발표를 담당하였습니다.

- <https://demoim.co.kr>
- <https://www.youtube.com/watch?v=fzlnCo4VSLE&t=2s>

😊 My Part

- Spring Security와 JWT토큰을 이용한 로그인 구현
- Http통신과 SetInterval을 통한 알람 기능 구현
- JPA JOIN 을 활용한 댓글 및 마이페이지 구현
- ACM, CloudFront, LoadBalancer를 통한 SSL-> HTTPS 구현

2021.04

Spring 기반의 Blog 개발

Spring 기본+심화 챕터에 걸친 OAuth2_카카오 로그인이 접목된 블로그 개발, 블로그의 Frontend 까지 직접 개발했습니다.

- https://github.com/strong1133/hh99_spring

😊 My Part

- Spring Security의 필터체인과 세션-쿠키를 이용해 로그인 구현
- OAuth2를 활용한 카카오 소셜 로그인 구현

2021.04

향해99 크루원들의 교류를 위한 APP, 크루톡의 Backend 개발

React Native와 Spring Boot로 만들어진 크루톡의 Backend 개발을 담당했습니다.

- <https://www.youtube.com/watch?v=x1XWfufVxCQ>
- https://github.com/strong1133/CrewTalk_BackEnd

😊 My Part

- Spring Security와 JWT를 통한 로그인 구현
- PageNation을 이용한 무한스크롤 구현
- JPA Containing을 사용한 검색 기능 구현

2021.03

환경운동 ZERO_Waste 을 홍보하기 위한 웹서비스 Backend 개발

Python Flask로 동작하는 Backend 서버와 Jinja 템플릿을 통한 SSR 웹페이지를 개발 하였습니다.

- https://github.com/strong1133/hh99_zero-waste

😊 My Part

- Selenium을 이용한 인스타그램 크롤러를 만들어 제로웨이스트 태그의 게시물을 크롤링하여 데이터 베이스 구축
- Flask를 활용한 서버 구축 및 JWT 토큰 방식의 로그인 구축
- MongoDB 사용을 위해 pymongo를 사용해 쿼리 생성

자기소개

원무과에서 개발자로

항상 메뉴얼대로만 하는 기계가 아닌, 새로운 생각을 하며 비전을 찾는 사람이 되고 싶었습니다. 병원에서 근무하는 기간동안 사실 참 재밌었습니다. 함께 일하는 사람들과 현 상황을 같이 이해하며 하루 하루를 해쳐나가는 것이 즐거웠습니다.

하지만 업무 자체는 정해진 메뉴얼대로 응대하고 상정되지 않은 특이사항 역시 메뉴얼을 강조하는 상사와 그안에서의 유도리는 저의 뚝이었습니다.

가치를 창출하지 않는 비전없는 삶에 한계를 느껴, 알을 깨고 새로운 세상으로 나아가고자 개발자로의 커리어 전환을 결심하였습니다.

뭐든지 경험해보자!

어려서부터 모든 것은 직접 경험해봐야 비로소 배움이 생긴다는 생각을 가지고 있었습니다. 해서 많은 아르바이트를 직접 해보았고 그안에서 배운 사회경험이 병원에서 일할때 아주 도움이 되었다고 생각합니다.

개발에서도 마찬가지입니다.

Backend를 공부하고 있지만 제가 만든 api를 Frontend에서 어떤식으로 요청하고 응답을 어떤식으로 View에서 활용하는지 직접 해보는 것 만큼 큰 배움은 없다고 생각합니다.

협업에서도 팀원이 짜놓은 코드를 항상 프로젝트 끝에 코드리뷰를 부탁하여 해당 코드에 담긴 의도를 파악하고 그것을 응용해서 새로 코드를 작성해 보는 등 하나하나 직접 경험해보고자 합니다.

또한 에러가 발생하였을때 다른 파트에서 발생한 에러일지라도 나서서 함께 해결하려 노력합니다. 왜 이런 에러가 발생했고 이것이 어떤 에러이며 그 해결 과정은 그 상황이 닥쳤을 때 경험해보는 것이 상당한 배움이 될것이라 생각하기 때문입니다.

몰두와 몰입

집중력이 좋은 편입니다. 어떠한 일에 잘 몰입 하는 편이며, 해당 업무가 완료 될때까지 몰두하겠습니다.

한가지에 깊이 파고 들어가보는 성격이며, 어떠한 이슈가 생기면 그 이슈가 해결 될때 까지 쉬지 않고 집중하겠습니다.

하지만 전체적인 진행과정을 항상 염두하여 전체적인 일정에 지장이 안가게끔 완급 조절에 노력하고 있으며 자칫 업무 흐름을 놓치는 것을 방지하기 위해 Task Block들을 Todo의 형식으로 문서화 해놓겠습니다.

깊게 탐구하는 것을 좋아하며 그 지식을 교류하고 또 지식이 더 깊은 사람의 지식을 배우는것을 좋아합니다.

자기소개

문제를 파악하고 구체적으로 해결한 경험

프로젝트 제로웨이스트를 개발할 당시 해당 프로젝트는 인스타그램 해쉬태그 게시물을 크롤링 해야했습니다. 인스타그램이 JS로 이루어져 있는 것을 발견하고는 셀레니움을 활용하여 크롤링을 시도 하였으나 인스타그램 특성상 게시물 탐색 중간에 로그인을 하라는 경고창이 떠서 크롤링 작업이 블락되는 문제가 있었습니다.

해당 문제를 해결하기 위해 최초 진입을 인스타그램 로그인 화면으로 하여 `find_elements_by_name("username")[0].send_keys("{ID}")` 등을 보내 로그인을 해주었고, `time.sleep(2)` 지연 후 실제 탄겟 url로 크롬 진입을 하여 최초에는 각 카드 게시물의 실제 url정보만 크롤링 하였습니다. 각 게시물이 모달이 아닌 페이지로써 접근하기 위함 이었으며, 게시물 실제 url정보량 만큼 for문을 돌려주며 각 게시물 페이지에서는 크롤링 하려는 데이터 요소를 BS4로 크롤링 하여 DB에 저장하여 해결 하였습니다.

스프링 프로젝트 당시에는 사용자 프로필 편집 기능에서 사용자가 정보를 한번 변경할 때 자기소개만을 바꿀 수도 있고 사진까지 같이 바꿀 수 있는 경우의 수가 있었습니다. 처음에는 사진을 바꾸는 api와 자기소개를 바꾸는 api를 따로 만들었으나 만약 사진과 자기소개를 동시에 바꾼다면 한 가지 요청에 두 api를 요청을 보내야 하는 문제가 있었습니다.

따라서 두 api를 하나로 합쳤으며 사진 파일은 form-data(multipart)의 형태로 body에 담아 요청을 받고 있었기 때문에 자기소개에 대한 항목 역시 form-data 문자열의 형태로 “{‘description’ : ‘안녕하세요’}” 이러한 형식으로 서버에 요청 하였고, 서버에서는 이를 `JSONObject jsonObject = new JSONObject(userEditInfo);`를 통해 JSON형식으로 DTO에 담아주는 방식으로 설계 하였습니다.

또한, 자기소개만 바꾸는 경우에는 `@RequestPart(required = false) MultipartFile file`를 통해 사진 첨부가 없으면 없는 대로 요청 수행이 가능하게끔 하였고 사진만 바꾸는 경우에는 클라이언트 측에서 리덕스의 현재값을 데이터로써 담아 넘겨 주게끔 하여 설계 하였습니다.

이력서 상의 모든 기재 사항은 사실과 다름이 없음을 확인합니다.

2021년 05월 31일
정석진