문성하

azqazq195@gmail.com

+82 1088977064 | 1994년생

개발 직무 서버/백엔드 개발자

기술 스택 Spring Boot, Kotlin, Java, AWS

학력

2020.02 졸업

대학교(4년) | **원광대학교**

컴퓨터공학과

경력(업무경험)

개발경력 1년 7개월

2022.10 - 2023.01

0년 4개월

㈜오늘의꽃

온라인 꽃 판매 (꽃 가게 및 일반 고객 대상)

개발팀/사원

NestJS, AWS, Kotlin, Spring Boot, GitHub Actions, TypeScript

- Legacy 코드 유지 보수 (TypeScript + NestJs)
  - Exception Handler, Validation, 인증/인가, Logging
- 신규 서버 설계 및 초기 인프라 구축 (Kotlin + Spring)
  - JPA, Github Actions CI/CD, AWS ECS
- 테스트 코드 작성 및 문서화

2021.07 - 2022.09

1년 3개월

코너스톤테크놀러지㈜

제조업 솔루션 (B2B)

개발팀/사원

Java, Spring, Mybatis, Jenkins

- BOM 솔루션 개발
- PG사 결제 및 세금계산서 API 연동으로 결제 자동화

- 도면파일 변환 로직 개선 (AutoLisp, ImageMagick)
- 테스트 자동화 구축 (Jenkins)

프로젝트

2023.04 - 2023.05

programmers helper

'프로그래머스'에서 코딩 테스트 문제를 intellij idea로 옮기는 plugin 입니다.

Kotlin | IntelliJ IDEA | Apache FreeMarker

- IntelliJ Platform SDK plugin 개발
- Jsoup 활용 문제 크롤링
- Freemarker 활용 언어별 템플릿 작성
- Github Actions을 통한 버전 관리 및 마켓 배포

웹/마켓/스토어

· https://plugins.jetbrains.com/plugin/21278-programmers-helper

저장소 링크

· https://github.com/azqazq195/programmers_helper

2022.03 - 2022.06

assistant

반복업무 단축을 위한 Flutter Window App

Flutter | Spring Boot | Java | Dart

- 코너스톤테크놀러지㈜ 에서 동료들에게 배포 이후 업무시간 단축
- svn에 등록된 ddl을 통한 mybatis 및 java class 파일 생성
- misx 패키지 및 개인서버를 구축하여 배포

저장소 링크

· https://github.com/azqazq195/assistant

교육이력

2020.06 - 2020.12

자바 안드로이드 웹&앱개발자

이젠 IT 아카데미

Java | Spring

- 웹 개발 프로그래밍 6개월 과정
- AWS 서버 구축 경험

- 팀 프로젝트 수행 경험
  - https://github.com/azqazq195/Project_Foret

자기소개서

실수를 방지하는 개발자

첫 직장에서 일을 경험하면서 업무 중 많은 실수를 저질렀습니다. 여러 가지 일 중에 한 가지에만 집중하거나, 빠르게 해결하고자 하는 의욕만 앞서 다를 것을 놓치는 경우가 많았습니다. 다음에는 실수하지 말아야지 하는 결심은 아무 도움이 되지 않는다는 것을 알게 되어 업무 일지를 기록하기 시작했습니다. 업무 시작 시 요구사항을 정리하고, 중간중간 확인해 가며 내가 설정한 방향성이 업무와 맞는지, 놓친 것은 없는지 파악되기 시작했습니다. 또한 이 일지를 통해 동료들과 피드백을 할 수 있었습니다.

전체적인 업무 이외에 개발할 때도 코드 스타일의 실수나 테스트를 돌려보지 않고 반영하는 일들이 있었습니다. 테스트 자동화를 통해 이러한 실수가 일어나지 않도록 방지하고, 추상화나 접근 제어 자를 통해 저뿐만 아닌 다른 개발자들도 실수하지 않도록 설계하여 방지하였습니다.

이러한 경험을 토대로 실수는 누구나 할 수 있고, 조심하는 것보단, 실수하지 못 하도록 설계 및 자동화를 통해 이를 방지해 가는 개발자로 성장하려 합니다.

반복업무를 싫어하는 개발자

어렸을 적 부터 반복적인 일을 좋아하지 않았습니다. 의미 없는 시간이라는 생각이 들고, 반복을 하다 보면 실수 등으로 결과물이 달라지는 경우가 있기 때문입니다. 개발하면서 높은 확률로 반복적인 일을 경험하는 것 같습니다. 그래서 반복적인 업무, 반복해야 할 것 같은 업무들은 자동화 시킬 방법을 먼저 고민하고 있습니다. 백엔드 개발자로서 서버의 CI/CD는 물론, ‘assistant’라는 사이드 프로젝트를 통해 반복적으로 생성하는 mybatis를 대신 생성해 주는 프로그램을 만들었습니다. 이를 동료들과 공유해 업무시간을 단축했던 경험이 있습니다. 여기서 얻은 시간으로 다른 업무를 일찍 시작하거나 더 좋은 설계를 고민할 수 있었습니다.

기록하는 개발자

실수를 방지하기 위해 개인 메모 수준으로 업무를 기록했었습니다. 이 메모들을 동료와 공유, 보고 등으로 활용하였고 이러한 방식이 효율적이라고 생각하게 되었습니다. 내 생각과 의도를 명확하게 전달할 수 있었고, 저와 동료들은 언제든지 열람할 수 있었고, 이에 대한 피드백도 받을 수 있었기 때문입니다. 이러한 경험으로 점점 문서의 형태를 갖추기 시작했습니다. 스타트업으로 이직하면서 개발자에게서 더 나아가 비 개발자와도 커뮤니케이션할 수 있는 자료가 되었고, 신입의 온보딩 과정에도 활용될 것을 기대하게 되었습니다. 기록에 대한 효율성을 경험으로 깨달았기 때문에 앞으로도 심도있는 개발 블로그나 공식 문서와 같은 글을 작성해보고 싶습니다.

링크

Github	**https://github.com/azqazq195**
BLOG	**https://moseoh.tistory.com/**

포트폴리오

목차

경력 기술

프로젝트

㈜오늘의꽃

2021-07 ~ 2022-09

Legacy 코드 유지 보수 (TypeScript + NestJs)

요구사항

- 디버깅 가능한 logging 추가
- 서버 문제 파악 및 수정

백오피스 사용 중 직원 중 한명이 데이터를 날리는 일이 있었습니다. 인가등의 권한 처리가 없었고 추적 가능한 logging이 없어 문제를 파악할 수 없었습니다. 유의미한 logging을 추가하고 추후 인가 개발을 위해 세션 정보를 활용할 수 있도록 수정하였습니다. 또한 온보딩 기간 서버 파악겸 개선사항에 대해 찾아보고 수정할 수 있는 시간을 가졌습니다.

진행내용

1. 디버깅 용 Logging 방식 변경
    - AS-IS: 무의미한 `"request"`, `"response"`, `"some method complete"` 등의 Logging만 존재
    - TO-BE: 응답 시간, 요청자, filter를 통한 error logging 등 활용가능하도록 수정
2. Error Filter 적용 (RestAdvice in Srping)
    - AS-IS: Exception Handler의 부재로 명시된 몇 에러 제외 Runtime 오류가 Client에 노출.
    - TO-BE: Exception Filter 구현
3. class-validation, class-transform 적용 (Validation in Spring)
    - AS-IS: Service 로직에서 일일히 입력검사
    - TO-BE: DTO에서 Decorator (Annotaion in Spring) 활용하여 중복코드 제거 및 책임 분리
4. ORM 설계

    - AS-IS: Typeorm을 사용하나 Object Relation을 활용하지 않음.
    - TO-BE: 신규 기능에 대해 객체 연관관계 설계 적용

5. Local 테스트 환경 구축

    - AS-IS: Local에서 테스트하기 위한 환경이 준비되어 있지 않음.
    - TO-BE: Docker Compose 를 통해 local db 환경 추가. (Github Actions 를 통한 테스트 자동화 구축 예정)

6. 요청자의 정보를 가져올 수 있도록 설계 (SecurityContextHolder.getContext().getAuthentication() in Spring)

    - AS-IS: Session에 User 정보가 있으나 활용하지 않아 client에서 매번 `user/me`를 호출하여 UserId를 추가함
    - TO-BE: 로그인된 사용자의 요청은 Session에서 UserId를 가져올 수 있도록 설계

신규 서버 설계 및 초기 인프라 구축 (Kotlin + Spring)

요구사항

- 기존 서버 대체, 신규 서버 설계

진행내용

1. 서버 구성

    - Kotlin, SpringBoot를 사용한 서버를 구성하였습니다.
    - JWT, JPA, QueryDsl, Redis, MySql, Restdocs

2. 테스트 환경 구성

    - local branch에서 유닛 테스트를 할 수 있도록 구성하였습니다.
    - develop branch에서 Github Actions를 통해 통합 테스트를 할 수 있도록 구성하였습니다.

3. 배포 환경 구성

    - preview, production branch는 각각의 VPC환경으로 배포하였습니다.
    - 배포에는 Github Actions, AWS ECR, AWS ECS를 사용하였습니다.
    - preview 환경은 production과 동일하게 구성하되, 비용을 고려하여 서버 사이즈 및 성능을 축소

코너스톤테크놀러지(주)

2021-07 ~ 2022-09

PG사 결제 및 세금계산서 API 연동으로 결제 자동화

요구사항

- 고객사 요금 계산 자동화
- 고객사 요금 청구 및 결제 자동화

고객사가 늘어남에 따라 엑셀로 일일히 계산하여 청구하는 것에 대한 시간이 증가하였습니다. 이 때문에 고객의 무료체험 전환부터 사용량 집계, 청구 까지의 프로세스에 대한 자동화의 필요성을 느껴 개발을 맡게 되었습니다.

진행내용

1. 고객의 무료체험 신청
    - 고객의 무료체험 시작 시 sales 담당자에게 메일로 알람을 발송합니다.
    - 담당자의 승인 후 고객사 용 서브도메인을 생성합니다. (수동)
    - 해당 고객사용 데이터 베이스를 생성합니다.
    - 무료체험 이후 사용을 종료하는 경우 한달의 유예기간 이후 데이터 베이스를 삭제합니다.
2. 라이선스 및 사용량 집계
    - 매일 새벽 고객이 사용한 라이선스 비용을 집계합니다.
        - 이는 고객이 라이선스를 자유롭게 바꿀 수 있기 때문에 매일 집계가 필요하였습니다.
    - 매일 새벽 고객이 사용한 파일 사용량(파일 업/다운로드, 도면 변환)을 집계합니다.
3. 결제 자동화
    - 집계된 금액을 결제 전 고객사에 청구합니다.
        - Freemarker를 활용한 청구서 메일 발송
    - 이체 고객의 경우 세금계산서를 발행합니다.
        - `PopBill` 서비스 연동
    - 자동 결제 고객의 경우 등록된 카드를 통해 결제를 진행합니다.
        - `Iamport` 서비스 연동

도면파일 변환 로직 개선 (AutoLisp, ImageMagick)

요구사항

- 도면파일 변환 다양화

고객사가 늘어남에 따라 도면 또한 다양해졌기 때문에 기존 고정된 크기로 변환만 시켜주는 로직의 개선이 필요하였습니다. 개선이 필요한 케이스는 아래와 같았습니다.

1. 도면의 표준 양식 제공
2. 표준 양식을 지켰다면 적절한 크기 변환하고 추가로 해당 파일을 읽어 도면의 프로세스 관리(승인자, 결제자, 날짜 등 스탬프)
3. 표준 양식을 지키지 않았다면 기본값 혹은 고객이 커스텀하게 변환
4. 도면 확대시 확대 부분만 재 변환

도면의 크기가 큰 경우가 가장 문제가 되는 부분이었습니다. dwg 파일로 열었을 경우 당연히 문제가 없고, pdf는 벡터 파일로 문제가 없었습니다. 하지만 png로 변환된 파일을 제공하기 때문에 크기가 큰 도면을 같은 사이즈로 변환하는 경우 파일의 크기가 커지고, 확대하지 않으면 잘 보이지 않는 등의 문제가 있었습니다. 때문에 확대를 하지 않는 기본값으로는 도면의 선들이 잘 보이도록 선명도를 조절하고, 확대를 하여 재랜더링을 요청하는 경우 해당 부분만 재생성하여 다시 잘 보이도록 구성하였습니다.

진행내용

1. 도면 변환 로직 개선
    - dwg -> pdf -> png 파일 변환
    - dwg -> pdf 에는 AutoCad의 AutoLisp 사용
    - pdf -> png 에는 ImageMagick 사용
    - png 변환 시에는 도면의 크기를 읽어서 변환, 기본값으로 변환 및 고객이 입력한 사이즈로 변환 옵션 제공
2. 도면 뷰어 개선
    - 도면에 필기 기록 저장
    - 도면 옵션(회전, 색반전 등) 저장
    - 확대 후 재랜더링 요청시 확대 배율로 변환

테스트 자동화 구축 (Jenkins)

요구사항

- 테스트 자동화 구축

신규 기능이 빠르게 추가되면서 테스트를 깜빡하거나, 유닛 테스트만 진행하여 다른 부분에서 새로운 오류가 발생하는 등의 문제가 있었습니다. 이 때문에 고객사들의 자사 솔루션 경험이 저하되었습니다. 이를 해결하기 위해 테스트 자동화 구축을 제안하고 진행하게 되었습니다.

진행내용

- 자사 local 서버에 Jenkins 구축

제한사항

해당 업무는 단순히 Jenkins를 도입하여 테스트만 자동화 하는 것에 그쳤습니다.

- branch는 하나만 사용
- 메이븐으로 구성되어있는 프로젝트이나, 모든 설정은 IDE(Eclipse)를 통해 진행되어 있음.
    - SVN에서 코드를 받아 그대로 빌드할 수 없어 이를 수정함 (설정, 라이브러리 등)
- 배포는 프론트 개발을 맡고 있는 협력 업체에서 진행

개인 프로젝트

아래 프로젝트는 개인으로 진행한 프로젝트 입니다. 해당 프로젝트들은 협력이나 성능 개선, 문제 추적 등을 어필기 보단 반복된 작업을 줄이기 위한 것에 초점이 맞춰져 있습니다.

Programmers Helper

프로그래머스 라는 사이트에서 코테를 자주 봤었기 때문에 취준기간 자주 사용하는 사이트 였습니다. Web IDE에서는 자동완성이 안되고 실행시간이 느리다는 단점으로 인해 외부 IDE에서 문제를 풀곤 했습니다. 하지만 문제를 풀때마다 코드를 옮기고 붙여넣는 과정이 낭비라고 생각되어 Intellij Plugin을 개발하게 되어습니다.

프로그램 기능

- 프로그래머스 사이트의 문제 url을 복사하여 패키지 및 파일 생성

- 파일에는 문제에서 제공되는 기본 코드와 테스트 케이스를 작성해 줍니다.
- 문제 풀이 이후 프로그래머스 사이트에 제출하기 위한 코드만 복사해 줍니다.

활용 기술

- Jsoup을 활용한 크롤링
    - 기본제공 코드 및 테스트 케이스를 읽어오기 위함입니다.
    - 프로그래머스 에서 API를 제공하지 않아 크롤링으로 진행되었습니다.
- Freemarker 활용 언어별 템플릿
    - 언어별로 테스트 케이스나 실행 코드가 다르기 때문에 Freemarker를 활용하였습니다.
    - 다른 언어를 개발할 경우를 대비해 확장성을 갖추기 위함입니다.
- Github Actions을 통한 CI/CD
    - 개발이후 release 버전 관리 및 Intellij MarketPlace 배포를 위한 자동화를 구현하였습니다.

Assistant

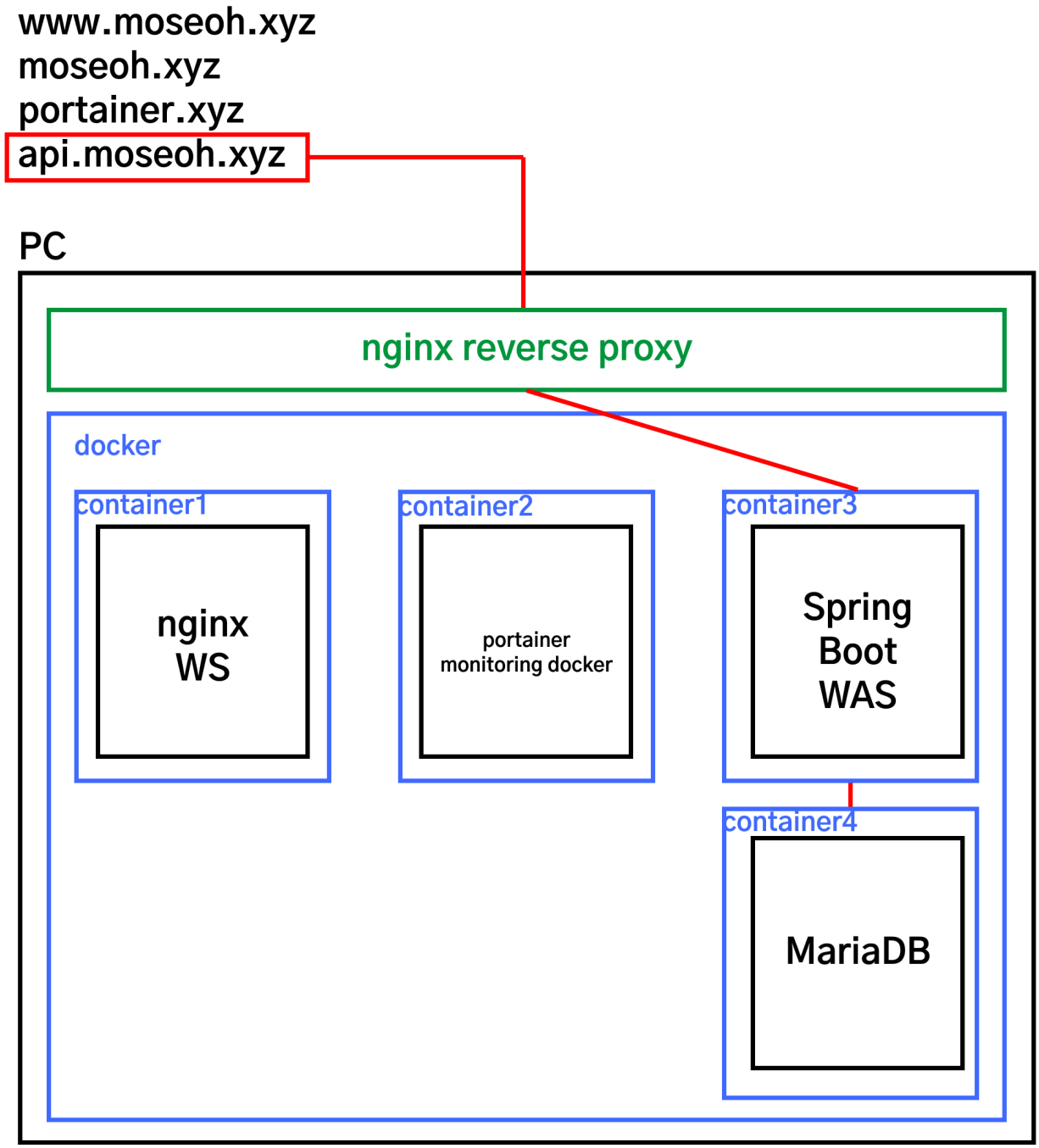
코너스톤테크놀러지(주) 에서 업무시간을 줄이기 위해 진행한 프로젝트입니다. 재직 당시 신규 기능 개발이 많았는데 매번 새로운 테이블에 대한 class, mapper, mybatis, test 를 만드는 시간이 불필요하게 느껴졌습니다. 해당 프로젝트를 통해 2시간씩 하던 작업을 5분으로 줄였습니다. 또한 동료들과 공유하여 좋은 평가를 받았습니다.

프로그램 기능

- SVN에 등록된 코드에서 ddl 관련 파일을 읽어드립니다.
- 작성하고자 하는 테이블을 선택하여 필요한 파일들을 만들어 냅니다.
- 해당 프로그램에 프로젝트 경로를 설정해두어 파일들을 적절한 위치에 저장합니다.
- 신규 테이블 뿐만아니라 기존 테이블의 수정사항도 작성해 주기 위해 diff 감지까지 개발을 진행하였습니다.

활용 기술

- Flutter을 활용한 Client 작성
    - Window App으로 배포하였습니다.
    - 파일을 생성하고 관리하길 원했기 때문에 Web보단 프로그램으로 작성하길 원했습니다.
- 서버 구성
    - DockerFile을 통해 image로 배포하였습니다.
    - AWS에 배포하지 않고 개인서버를 구축하였습니다. (비용 부담 및 연습용)
    - Container 활용, 도메인 호스팅


www.moseoh.xyz
moseoh.xyz
portainer.xyz
api.moseoh.xyz
PC
nginx reverse proxy
docker
container1
nginx
WS
container2
portainer
monitoring docker
container3
Spring
Boot
WAS
container4
MariaDB