| 송지민  Backend Engineer  Email : jimin.song.1999@gmail.com  Phone : 010-9171-0281  GitHub :<https://github.com/jiminsong490> | |  |
| --- | --- | --- |

**경험**

| **아르테일 (Artale) /** Backend, DevOps **메이플스토리 월드 게임** | | 2023.10~2024.07  World Play : [Artale](https://maplestoryworlds.nexon.com/ko/play/1907c8d7a23f4742a616b59ac752d7e8/shop) |
| --- | --- | --- |

**기술 스택**

* Node.js, NestJS, TypeORM, MySQL, MongoDB, React
* AWS (Elastic Beanstalk, Lambda, SQS, EC2, RDS, CloudWatch), Docker

**기여한 일**

* 기존 유저 간 1대1 거래 기능을 개선한 **서버 간 통합 아이템 거래 서비스** 개발 ([**링크**](https://artale.one/posts/%EC%95%84%EB%A5%B4%ED%85%8C%EC%9D%BC%EC%9D%84-%EC%A6%90%EA%B8%B0%EA%B8%B0-%EC%9C%84%ED%95%9C-tip-%F0%9F%92%81%E2%80%8D%E2%99%82/%EC%95%84%EB%A5%B4%ED%85%8C%EC%9D%BC-%EC%98%A5%EC%85%98%EC%9D%B4-%EC%9E%88%EB%8B%A4%EB%8A%94%EB%8D%B0-%EC%98%A5%EC%85%98%EC%9D%B4-%EB%AD%94%EA%B0%80%EC%9A%94-1353134?search=&page=&subject=))
  + 아이템 검색, 등록, 만료, 구매, 수령, 등록 취소 기능을 **REST API**로 개발
  + Elastic Beanstalk을 통해 배포하여 **Auto Scaling** 및 **Load Balancing**을 적용
  + 서버의 상태를 지속적으로 확인하여 컨테이너가 다운됐을 경우 자동으로 재시작 하도록 설정
  + 아이템 구매 시 발생할 수 있는 **동시성 문제**를 비관적 쓰기 락 설정으로 해결
  + 각 기능들이 하나의 작업으로 동작하게 하기 위해 createQueryRunner로 **트랜잭션** 처리
* **100여대의 서버**에서 유저들이 생성하는 로그 데이터를누락없이 안전하게 저장
  + API Gateway > SQS > Lambda > MongoDB로 이어지는 **로그 저장 인프라 구축**
  + Lambda를 **Consumer**와 **Worker**로 나누어 20여가지가 넘는 로그 데이터를 **분산 처리**
  + **초당 8천개** 이상의 요청을 안정적으로 처리하기 위해 SQS로 **요청 대기열 구성**
  + **DLQ**를 생성하여 요청에 실패한 로그들을 관리하고 누락이 없도록 관리
  + UTC시간을 기준으로 매 00시에 전날 저장된 데이터를 CSV파일로 변환하여 S3에 백업
* 저장되어 있는 로그 데이터를검색할 수 있는 **관리자 사이트**를 개발 및 운영
  + 유저 아이디를 통해 특정 유저가 생성했던 로그들을 모아 볼 수 있는 관리자 사이트를 개발
  + MongoDB의 Aggregation으로 **저장된 로그들을 집계**하여 그래프용 데이터로 사용
  + 로그의 수많은 필드 중 원하는 필드만 가져올 수 있도록 **GraphQL**로 API를 구현
* 매일 변경되는 유저들의 아바타를 갱신하기 위한 **크롤링** **작업** 개발
  + EventBridge + Lambda + Selenium + ECR을 사용하여 매일 8시 크롤링 진행
  + 실행 환경에 영향을 많이 받는 Selenium의 특성을 고려하여 Docker를 사용

**프로젝트**

| **Check-it /** Backend, DevOps **데일리 독서 기록 서비스** | | 2023.03 ~ 2023.08  Github : [Check–it](https://github.com/techeer-sv/Checkit) |
| --- | --- | --- |

**기술 스택**

* Spring Boot, JPA, Spring Batch, MySQL, MongoDB, Redis
* AWS (EC2, RDS), Docker, Github Actions

**기여한 일**

* **책, 독서 정보** 도메인 RESTful API 개발
* **Spring Batch**를 사용하여 크롤링 데이터를 MongoDB와 MySQL, Redis에 저장 ([**링크**](https://jiminsong490.github.io/posts/SpirngBatch+%ED%81%AC%EB%A1%A4%EB%A7%81/))
  + **TransactionManager**를 사용하여 step이 실패할 경우 원래 상태로 **rollback**
  + 크롤링 시 Redis를 확인해 이미 크롤링 된 책인지 확인하는 방식을 통해 **데이터 중복을 예방**
* **Github Actions**로 지속적인 통합(**CI**), 지속적인 배포(**CD**) 파이프라인 구축 ([**링크**](https://jiminsong490.github.io/posts/Docker_hub+Github_Actions(CICD)/))
  + Pull Request 시 테스트와 빌드를 진행하고 통과한 경우에만 merge 허용 (CI)
  + merge 후 변경된 이미지를 빌드하고 **Docker Hub**에 배포, **Jib**로 이미지 빌드 시간 단축
  + SSH를 통해 EC2에 접속하여 배포된 이미지를 Pull한 후 컨테이너를 실행

| **MoaMoa /** Backend | **사이트 프로젝트 팀원 매칭 서비스** | 2023.03~2023.07  Github : [MoaMoa](https://github.com/Team13-MoaMoa) |
| --- | --- | --- |

**기술 스택**

* Spring Boot, JPA, MySQL, Swagger, RestDocs, Docker

**기여한 일**

* **유저, 게시물, 댓글, 쪽지** 도메인에 대한 REST API 개발
  + **QueryDSL**을 사 용하여 참여 포지션, 사용언어, 검색어 유무에 따른 **동적 쿼리 구현**
* **Swagger**와 **RestDocs**의 장점만 적용한 API 명세서 작성 ([**링크**](https://jiminsong490.github.io/posts/RestDocs+Swagger/))
  + OAS 기반의 Swagger와 RestDocs를 동일한 API spec 파일로 명세하도록 설정
  + 테스트 코드 강제를 통한 코드 안전성 확보와 Swagger의 편의성을 동시에 제공

**사용 기술**

| * **Language** : Node.js, Java * **Framework** : NestJS, Spring Boot * **ORM** : TypeORM, JPA | * **Database** : MySQL, MongoDB, Redis * **AWS** : Lambda, EB, EC2, RDS, CloudWatch * **DevOps** : Docker, Nginx, Github Actions |
| --- | --- |

**학력**

| 성결대학교 /2018.03 ~ 2024.02 졸업 컴퓨터 공학부 | | |
| --- | --- | --- |