

100세까지 개발로 세상에 기여하고 싶은 개발자



설 훈 (Hun, Seol),
“나라에 향기나는 사람이 되라”

Mobile : 010-2902-4829

E-Mail : shun10114@gmail.com

Github : <https://github.com/Seolhun>

Blog : <https://seolhun.github.io/>

간략한 소개

어렸을 때부터 레고부터 장난감 분해/조립, 요리까지 만드는 것을 상당히 좋아했습니다. 창조하는 것을 좋아하여 새로운 아이디어를 기획하기 시작했고, 이를 개발하기까지 이르게 되었습니다. 개인적으로는 플랫폼 개발과 로그/검색/추천 서비스에 관심이 많으며, 개발자로서 이를 잘 만드는 것이 10년 안의 목표입니다.

현재 스타트업에서 시작하여 약 3년차가 되어가고 있습니다. 제가 좋아하는 것은 세상에 이로운 서비스를 만들기 위해 프로세스를 정립하고, 동료들과 열린 토론을 할 수 있는 개발 문화라고 할 수 있습니다. 그렇다면, 전 어디서든 일할 수 있습니다.

코드를 합리적으로 효율화 할 수 있는 새로운 기술을 좋아하고, 처음 접한 기술로 서비스를 만드는 도전적인 일도 좋아합니다. 문서 읽는 것을 좋아하고 개발하면서 이를 정리하여 공유하는 것도 좋아합니다.

보유 기술

- Programming Language (배점 1 ~ 5점)

NodeJS (3), Java (2~3), Python (1)

JavaScript(3~4), TypeScript (3)

HTML5, CSS(SASS) (3)

- Framework / Library

Express (3), Spring (3), Flask (1), Serverless (1)

Webpack(3), Lerna(2)

React (4), Vue (2), Mithril (2), Bootstrap (2), jQuery(2)

Styled-Component(2)

- Server

Tomcat(1), NginX(1),

- Environment

AWS (1~2) - (EC2, Lambda, RDS, S3, API GateWay, CloudFormation)

Linux(2), MacOS

- Tooling / DevOps

Github(3~4), CircleCI(2), Jenkins(2), Docker(1)

- Etc

JIRA, Waffle.IO, Slack, Codacy, ZenHub

프로젝트

2018.5 ~

ab180

프론트엔드 개발

Airbridge Dashboard 개발/개선

- Router Migration & Refactoring
 - 프로젝트의 구조화가 전혀되어있지 않은 문제 해결
 - 3 * 3 계층으로 나누어 HoC로 이를 처리
 - 1 Layer : Authentication, 2 Layer : AppList, 3 Layer : Dashboard
 - 각 Layer 이동마다 HoC 패턴으로 유효성을 검사하여 어떠한 페이지를 렌더링 할 지를 처리합니다.(에러처리 포함)
- Redux Refactoring
 - Thunk를 통한 비동기처리 이슈가 심하여 Observable로 개선
 - 그 외 Global 큐를 이용한 문제 등 다양한 문제 개선
 - 내용이 많아 질문주시면 답변드리겠습니다.
- HelperUtils Refactoring
 - 사용되지 않은 코드 및 사용할 수 없는 코드 모두 개선
 - Monad Pattern을 통한 Funtional한 코드로 변경
 - lodash를 이용하여 우리 코드 형태로 Wrapping
- FormBuilder Function 개발
 - Form을 자동화할 수 있는 코드가 없어 이를 구현
 - JS Monad 형식으로 구현하여 Framework에 상관없이 적용가능
- TimeLag 개발
 - CTIT 등의 Fraud 감시를 위한 Graph, Table View 작업
 - Modal > Drawer로 Configuration 추가 작업
- Dashboard Table 개선/개발
 - Multi Header, Freezing 기능 추가
 - Schema 구성을 통해 Props로 Table 자동화
- CircleCI workflow 도입
 - Branch에 따른 flow를 두어서 이를 처리하기 위함
- Storybook, Cypress 도입
 - Design UI 공유 및 End to End 테스트를 위함
 - 기본 작성방법 등에 대한 문서작업

프로젝트

2017.12 ~ 2018.03

(주)크로키닷컴

인턴 / 백엔드 개발

지그재그 MSA 서비스 개발/개선

- 지그재그(Seller, Management) Admin 서버/클라이언트 개선
 - Thrift(JS) REST > GraphQL Interface로 변경/추가/개선
 - CoffeeScript > TypeScript로 추가/개선
- Jenkins Pipeline을 통한 테스트케이스 추가
 - Storybook UI 테스트기능 추가(S3에 배포하여 Github Notification 기능 제공)
- 지그재그 스크레이퍼 작성 및 문서화 추가/개선
 - 지그재그 스크레이퍼의 부족한 문서화 작업 및 스크레이퍼 작성

프로젝트

2017.04 ~ 2017.09

(주)아이메디신

연구원 / 백엔드 개발

치매예방 뇌파 분석 플랫폼(iSyncBrain) 개발

- 보고서 Module 개발(차트, PDF변환 등) - 뇌파, 설문에 사용
 - JavaScript Module Pattern을 적용하여 재사용할 수 있는 보고서 Module 개발
- 사용자 설문을 위한 설문 Module 개발(Google Forms같은 설문)

- jQuery와 VueJS를 사용하여 설문 작성/제출/분석 서비스 개발
- 기존 JavaScript 코드 분리/모듈화
 - 기존 HTML과 같이 존재하는 JS파일 분리 및 Module화 적용
- On-Premise 서버 관리 및 NginX 설정
 - 홈페이지와 플랫폼 WebServer 구축 및 Https 설정 완료
- 프로젝트 관리를 위한 RedMine 설치 및 의사소통을 위한 Slack 제안
- CTO님을 도와 TIPS선정 및 SEED 투자 유치 성공
 - 뇌파 보고서 자동화 플랫폼 개발

프로젝트

2016.12 ~ 2017.03
(주)아이메디신
연구원 / 백엔드 개발

치매예방 뇌파 분석 엔진 데이터베이스 설계

- Matlab으로 작성된 뇌파 분석 알고리즘 분석하여 데이터베이스 설계
 - 알고리즘 분석통해 필요한 데이터베이스 구축
- Python Engine의 분석함수 Pipeline 구축
 - JNI를 이용한 Python 함수 호출 Pipeline 구축
- Jenkins를 사용한 CI 구축
 - 소스코드 안정화를 위한 Jenkins 테스트/빌드까지 구축
- Nas서버를 통한 회사 데이터 백업관리
 - Synology Nas를 통한 회사 컴퓨터 백업 및 서버 백업 구축

개인 프로젝트

2018.11 ~

(Main) Localize-Component 개발

- 현재 운영되는 Product에 SideEffect없이 도입되는 Component 개발이 목표
 - React, TypeScript, Emotion, Webpack를 통한 localized component 개발
 - Lerna를 통한 Mono Repo형식으로 구현
 - 내부 node_modules의 파편화를 방지하고 서로의 모듈을 참조한 개발가능
 - 변경사항에 대한 Npm에 배포편리
 - CircleCI를 통한 문서와 예제관련 사항 자동 프로세스화
- <https://github.com/Seolhun/localize-components>

개인 프로젝트

2018.04 ~

(Main) Hi-Cord Cralwer 개발

- Tech 관련 개발자들의 유익한 글들을 수집하기 위해서 개발하게 되었습니다.
- NodeJS, CheerIO로 구현한 Cralwer 개발
- React/carlo를 이용한 DeskTop app 개발
- Lerna를 통한 Mono Repo형식으로 구현

개인 프로젝트

2018.04 ~

(Main) Hi-Cord 개발

- Medium같은 개발자를 위한 글을 관리하고 추천/검색하는 서비스를 목표로 합니다.
- NodeJS, Express, Java, Spring, Serverless로 API 및 서버구현
- React, TypeScript,를 이용한 SPA 개발
 - Service, Repository Layer를 항상 분리하려고 노력
- Lerna를 통한 Mono Repo형식으로 구현
- Jest를 이용한 TestCode 작성

개인 프로젝트

2017.11 ~

(Main) 매일 개발관련 기록을 정리하는 회고 프로젝트입니다.

- Java, Python, JavaScript, Vue, 등 학습내용 정리.
 - 문법, 알고리즘, 개념 등을 주로 정리합니다.
 - 해당 내용들은 개선하여 블로그로 공유합니다.
- <https://github.com/Seolhun/retrospective-diary>

개인 프로젝트

2017.10 ~

회사에서 사용할 수 있는 책 관리 웹 프로젝트

- Vue, TypeScript를 이용한 SPA 개발
 - NodeJS, Express로 서버구현
 - API Interface로 GraphQL 구현
 - Repository, Service Layer로 분리
 - Jest를 이용한 TestCode 작성
- <https://github.com/Seolhun/vue-type-graphql-example>

팀 프로젝트

2016.04 ~ 2016.05

Spring MVC1을 사용한 일정관리 플랫폼

- Ajax를 사용한 비동기통신 구현
 - Oracle 10g로 일정관리 데이터베이스 설계
 - SQL Join 통한 각 일정검색 기능 구현
 - 기본 Session을 통한 회원가입, 로그인 구성
- <https://github.com/Seolhun/sist-second-project>

팀 프로젝트

2016.05~ 2016.06

Spring MVC2와 NoSQL을 사용한 음악 플랫폼

- Jsoup을 이용한 스크레이퍼 구현
 - Bugs, Mnet 등의 데이터 스크랩
- MongoDB를 사용하여 NoSQL 경험
- Google YouTube API를 사용하여 차트 노래리스트 구현
- Spring Tiles로 Layout 구성
- Google, Facebook OAuth2를 이용한 회원가입, 로그인

<https://github.com/Seolhun/sist-third-project>

팀 프로젝트

2016.04 ~ 2016.05

Spring MVC1을 사용한 일정관리 플랫폼

- Ajax를 사용한 비동기통신 구현
- Oracle 10g로 일정관리 데이터베이스 설계
- SQL Join 통한 각 일정검색 기능 구현
- 기본 Session을 통한 회원가입, 로그인 구성

<https://github.com/Seolhun/sist-second-project>

팀 프로젝트

2016.01 ~ 2016.03

Java Socket을 사용한 달무리 카드 게임

- Java Socket을 이용한 채팅구현
- Java Swing을 이용한 Client 구성
- Oracle 10g를 이용하여 회원가입, 로그인 구성
 - JDBC Connector사용
- Server Protocol을 정의하여 Thread로 게임 구현
- 5인 게임 및 대기실을 통한 최대 20명 정도 게임 가능.

<https://github.com/Seolhun/sist-first-project>

개선/문제해결 사례

(React/JS) Re-Rendering 문제, Router의 Migration 및 구조 해결 사례

1. 문제

- Router의 Hierarchy 문제
- Router Component의 Re-Rendering 문제
- Alert Hell. Redirect Hell

2. 원인

- 로그인 > 앱 > 대시보드 구조로 계층 별 필요한 권한과 필요한 데이터를 사전에 확인하지 않음
- Layout에 있는 값이 가장 먼저 변이되고 난 뒤에 Route의 값이 변이되지 않은 문제
- Landing에서 SignUp/In을 해결하고 이를 Airbridge로 redirect 시키는 문제.

3. 측정

- Layout과 Router Component과 함께 랜더링 될 때 Re-rendering이 최대 3~4번 발생
- 유효하지 않은 유저가 Dashboard 밖에 Homepage로 redirect되는 문제
- Alert가 최대 5~6번 뜨는 문제

4. 해결책

- Router의 버전업을 통한 Router의 Dynamic Change로직 추가
- 각각의 View에 대한 우선순위를 통해 사전에 필수적인 값이 없을 경우에 따른 처리 다변화
- Authentication 관련된 사항을 Dashboard로 Migration

5. 해결과정

- Router의 버전업을 통한 Router 관리 코드 개선
- HoC 함수 작성(withAuthentication, withDashboard)를 통해 User, App의 유효성 검사
- Authentication 관련 사항 Migration 및 3 * 3 Router Layer를 통해 해결.

6. 평가

- Alert 처리제거 및 불친절한 Redirect 문제해결 및 비즈니스 로직에 맞는 에러처리 효과
- 각각의 View(Container) 내에서 사전 데이터 체크 로직이 제거하는 긍정적인 효과
- 3~4회 Re-rendering이 일어나는 문제를, 1회만 랜더링만 되었습니다.

7. 비고

- 사전에 처리해야 할 데이터가 내에서 변경될 경우(자기 정보 변경 등)에 대한 처리에 대한
Action이 추가해야하는 이슈가 있었습니다. (Spinner관련 문제로 인함)
- 이번 작업은 Issue, PR, 구조 문서화 작업까지 병행하면서 하느라 많은 힘이 들었지만,
이러한 작업이 Over Engineer를 해결하는데 많은 도움이 되었습니다.
- Router는 Client Side의 중요한 부분을 차지하며, 버전업에 따른 변경코드 작업이 상당히 많은 부분이 힘들었으며, 코드 리베이스가 계속 진행되어야 하므로 많은 어려움이 있었습니다.

개선/문제해결 사례

(Java) 자바 객체 타입 문제로 인한 도메인 설계 개선 사례

1. 문제

- 코드 작성 간 @SuppressWarnings으로 경고 억제
- 객체 값 확인의 어려움과 타입 캐스팅 문제 발생

2. 원인

- 기존 코드에 객체 타입을 Map에 Object로 정의하여 타입 불완전성으로 인한 문제

3. 측정

- 리스트에 리스트처럼 Depth가 깊어질수록 해당 타입을 확인하고 이를 캐스팅 하는 로직이 많이 들어가는 현상 발생

4. 해결책

- 문서화 혹은 각 객체 Class를 구현하여 타입을 지정하는 방법.

5. 해결과정

- 문서화는 변경 시 작업량을 늘릴 뿐 해결책이 되지 않으므로 직접 Domain Class를 구현하여 코드 개선

6. 평가

- 타입 및 값을 확인하는 로직이 줄어들어 로직 별 코드가 최대 4줄이 줄었습니다.
- @SuppressWarnings를 사용하지 않아 타입의 안정성을 얻을 수 있었습니다.

7. 비교

- Domain을 설계함으로써, 객체 생성과 함께 필요한 기능을 Domain에서 직접 사용할 수 있어, 추가 로직을 작성하지 않아도 문제를 해결할 수 있는 기회가 확대되었습니다.

개선/문제해결 사례

(JS) 자바스크립트 중복 함수 제거를 위한 모듈 개선 사례

1. 문제

- JavaScript 소스 코드 작성 간 변수 값의 변경 및 충돌 문제 발생

2. 원인

- HTML 안에 JS 코드가 함께 존재하여 변수 변환 및 네이밍 충돌 발생

3. 측정

- HTML 안에 JS코드가 공통 JS와 충돌 및 중첩되어 디버깅의 어려움 발생
- HTML 별 작성되어 중복코드가 증가되고 같은 로직에 네이밍 변경되는 문제 발생

4. 해결책

- JS파일로 코드를 분리하고 Module Pattern 및 Prototype을 사용하여 함수를 재사용 할 수 있도록 구성

5. 해결과정

- 은닉화가 필요할 경우 Module Pattern을 적용, 그 외에 경우 Prototype으로 구현.

6. 평가

- JS파일로 분리하여 공통 로직을 하나의 module로 줄이고 필요한 경우에 따라 파일을 import할 수 있게 해결.

7. 비교

- JavaScript의 함수를 Class처럼 사용할 수 있었으며, 해당 함수를 변수에 받아 사용할 수 있게 만듦으로서 유지보수 및 렌더링 속도 개선

1 개발을 왜 시작하게 되었는가?

[표현의 욕구]

저는 머리에 있는 생각들을 직접 만들고 표현하는 것을 상당히 좋아합니다. 개발을 시작하는데 생각을 정리하고 표현하기 위해 공모전을 시작하게 되었고 이것이 개발에 관심을 갖게되는 계기가 되었습니다.

2011년 1학기 대학교 교내 공모전에 '핀테크:미래 결제기술 변화'라는 내용으로 참여하여 본선에 진출하였습니다. 기술적인 정보를 많이 얻었고 개발에 관해 관심을 갖게되는 계기가 되었습니다. 그 후에도 설빙 공모전에서 장려상을 수상받았습니다. 특히, 경산 콘텐츠 쇼에 3일 간 참여하여 개발자들을 만나 많은 얘기를 나누었습니다. 그리고, 표현을 더 구체화하기 위해 개발을 시작해야겠다는 결심을 하게되었습니다.

개발을 본격적으로 시작한 것은 전역 후 2016년 9월입니다. 원래 데이터에 관심이 많아 금융권 IT를 목표로 시작하였으며, 개발학습은 주로 독서와 강의 등을 들으면서 시작하였습니다. 이후엔 체계적인 커리큘럼과 팀 프로젝트를 느껴보고 싶어 학원에서 자바 기반 빅데이터 과정을 수강했습니다. 부족한 시간을 보충하기 위해 6개월간 자취를 결심했고, 스터디 그룹을 만들어 프로젝트를 진행하였으며 공부한 것을 나누고 정리하는 시간을 가졌습니다. 이 스터디를 통해 정보처리기사도 취득할 수 있었습니다. 이후 스타트업에 입사하여 치매 예방 뇌파 플랫폼 개발에 업무를 시작하였습니다. 뇌파 데이터 보고서와 설문 기능을 담당하였고 성공적으로 마무리할 수 있었습니다. 2017년 3월부터는 고려 사이버대학교 소프트웨어공학과에 편입하여 컴퓨터 사이언스에 대한 지식을 더욱 함양하기 위해 노력하고 있습니다.

[세상을 연결하는 개발자]

개발을 시작할 때의 스스로 정한 단기간 목표는 플랫폼을 개발하는 것입니다. 플랫폼을 개발을 통해 세상을 좀 더 편하고 이롭게 연결하고싶었기 때문입니다.

기술적으로는 데이터에 관심이 많아, 플랫폼을 통해 로그 데이터를 수집/분석하여 검색 및 추천 서비스를 개발하고 싶습니다. 특히, 로그서비스에 관심이 많습니다. 무엇인가 관찰하고 이를 응용하여 가치를 만들어내는 것을 좋아합니다. 또한, 무엇인가를 추론하고 증명하고 싶어하는 욕구가 있습니다. 로그 서비스와 검색/추천 서비스 개발을 다음 커리어패스로 나아가고 싶습니다.

이를 위해 개인적으로는 Github 기반 SNS 플랫폼을 만들고 있습니다. 이 개발을 통해 로그 데이터 분석 및 검색/추천 서비스 개발의 경험을 느껴보고자 노력하고 있습니다.

2. 개발자로서 성격의 장단점

[끈기와 집중력]

전 회사 CTO님과 팀장님이 "끈기와 집중력"이 강하다고 말씀해주셨습니다. 전 문제를 해결하기 위해 끊임없이 노력하는 스타일입니다. 문제를 해결 하기 위해 개발 외 시간에는 문서와 강의, 독서 등을 보며 부족한 것을 습득합니다. 습득한 내용은 간단히 테스트하여 Hello World라도 확인할 수 있게 응용하려고 노력합니다. 그리고, 글을 쓰면 본인이 알고 있는 것을 더 생각하고 정리할 수 있기에 글을 쓰려고 노력합니다.

[서툴지만, 프로세스화]

이러한 노력에서 서툰 모습을 보여줄 때가 있습니다. 서툰 모습을 제거하기 위해 프로토콜처럼 스스로에게 약속을 세워 지키려고 노력합니다. 특히, 깃허브 커밋방법, 글 쓰기 양식 등 다양한 부분에서 규칙을 세워 이를 개선하고 있습니다.

구조적인 자동화와 체크리스트를 통해서 이 문제를 해결하려고 노력합니다. 그리고, 체크리스트와 자동화를 주기적으로 업데이트하면 해당 문제를 더 구조적으로 파악하기 쉬워 좋습니다.

3. 개발 경험, 자세한 이야기

AB180의 Dashboard를 개발하였습니다.

먼저, Router의 구조가 재랜더링 및 구조적이지 않은 문제가 있어 이를 구조화하고 비즈니스 로직을 반영할 수 있도록 처리하였습니다. Action은 단일 Action 구조에 Sequence처리가 전역 Queue로만 처리하여 복잡한 비동기로직을 단일 Transaction으로 처리 할 수 없는 이슈가 있어, Middleware Redux Observable 도입하여 해결했습니다. 그외 많은 Action구조 함수를 통해 구조화하였습니다. 마지막으로, HelperUtils같은 것이 제대로 구현되지 않아 이를 개선하였습니다. lodash의 도움을 받았지만, Native 자바스크립트화 호환 될 수 있도록 구현하였으며, 함수의 재호출의 불편함을 해결하기 위해 Monad 형식의 Functional Code로 개선시켰습니다.

최근 크로키 닷컴의 지그재그 관리자 서버를 개발하였습니다. 먼저, 크롤링을 추가개발 및 유지보수 하였고, TypeScript와 GraphQL로 마이그레이션 중이라 해당 서비스를 마이그레이션하는데 많이 참여하였으며, 마케팅 대행사 가입 서비스를 만들기도 하였습니다. 추가적으로는 Jenkins Pipeline을 이용하여 Storybook UI 테스트 로직을 추가하였습니다.

두 번째, 아이메디신 뇌파 플랫폼을 개발하였습니다. 뇌파 플랫폼 개발에서는 요구사항 분석부터 설계까지 같이 참여할 기회를 얻어 개발할 수 있었습니다. 다양한 의견을 듣고, 얘기할 좋은 기회가 되었습니다. 주로 맡은 업무는 뇌파 보고서와 설문 서비스입니다. 뇌파는 Python-Engine을 통해 분석되어 해당 데이터를 보고로 만들 수 있도록 개발하였습니다. 설문은 Google Forms처럼 설문을 만들고 제출하여 결과를 통계 낼 수 있도록 만들었습니다. 이 외에도 On-Premise 서버로서 NginX, Tomcat, Jenkins 등 다양한 설정을 직접 경험하였습니다.

세 번째, 일정관리 웹 프로그래밍과 음악 빅데이터 프로젝트를 진행하였습니다. MVC1, MVC2를 모두 경험할 수 있었습니다. 특히, Ajax를 처음 사용하여 일정관리를 비동기적으로 움직이는 프로그래밍을 통해 비동기 통신을 이해할 수 있었습니다. 음악 빅데이터 프로그래밍을 통해서 크롤링과 NoSQL 서버를 이용하여 Bugs, Mnet 등의 데이터를 분석하여 감정에 맞는 음악을 추천할 수 있게 만들었습니다. 지금 생각하면 빅데이터라고 할 수 없지만, 새로운 스택을 이용한 흥미로운 개발이었습니다.

마지막으로, 처음 개발했던 것은 Java를 이용하여 Dalmuti라는 카드게임을 만든 것입니다. Java Socket과 Thread, FX를 이용하여 개발하였고 대기실과 방을 구현하여 5명이 같은 방에서 게임을 할 수 있게 만들었습니다. 이 프로젝트에서 가장 큰 배움은 Socket 통신의 개념을 체험하고 이해할 수 있었습니다. 저희가 정의한 프로토콜에 의해 서버가 작동된다는 것이 제일 흥미로웠습니다.

적은 시간에도 다양한 프로젝트를 경험한 시간이었습니다. 현재는 프로젝트를 통한 개발보다 개념에 대한 정리와 이해를 우선으로 집중하고 있습니다. 서버 네트워크와 알고리즘, 클린 코드 등 다양한 책들과 학습을 통해 진행하고 있습니다. 관련된 사항은 Git hub을 통해서도 확인할 수 있습니다.

학력 사항	<ul style="list-style-type: none"> - 1990.01.26 (양력) 남 - 용인대학교 (2009.03 ~ 2013.02) 스포츠미디어, 경영학과 졸업 - 고려사이버대학교 (2017.03 ~ 2020.06) 소프트웨어공학과 편입
병역 사항	- 군필 (2013.03 ~ 2015.06), ROTC 51기
자격 사항	- 정보처리기사 취득(2016.05.17)
외국어	- 오픽 IM2 (2017.02.22)
재미로 보는 기타사항	<ul style="list-style-type: none"> - 2011년 한국지도자 육성 장학재단 41기 장학생 및 총무 - 2016년 설빙 공모전 장려상 - 2017년 7월 ~ 8월 마이다스아이티 웹 기획 인턴 - 간단한 체육 및 금융 자격증 보유

여기까지 페이지를 넘겨주셔서 감사합니다.
더 궁금한 내용이 있으시면
shun10114@gmail.com으로 연락주세요.