

# Choi Sohyun

☎ 010-6676-5567

❤️ [sohyun4283@naver.com](mailto:sohyun4283@naver.com)

🐱 [github.com/choisohyun](https://github.com/choisohyun)

📝 [egg-programmer.tistory.com](https://egg-programmer.tistory.com)

## Profile

공유하고 함께 고민하면서 성장하는 프론트 엔드 개발자입니다.  
배운 것을 블로그에 포스팅하는 것과 기술 블로그 아티클 읽는 것을 좋아합니다.  
웹 사이트의 성능 개선 방법에 관심을 가지고 공부하고 있습니다.  
주 사용 프레임워크는 React입니다.

## Education

서울여자대학교 컴퓨터학과 – 졸업, 2015.03 ~ 2019.02  
청년 AI·Big Data 아카데미 7기 – 수료, 2019.07 ~ 2019.09  
코드스쿼드 마스터즈 과정 – 수료, 2020.01 ~ 2020.06

## Main Project

숙소 예약 서비스 – 2020.05 ~ 2020.06

Stack – react, react-dom, react-redux, babel, styled-component

숙소 데이터는 한 번에 1,000개 이상이 있었고, 데이터를 모두 불러올 때까지 3초가 넘는 시간이 걸렸습니다. 이런 문제를 해결하기 위해 무한 스크롤(Infinite Scroll)을 자바스크립트 내장 API인 Intersection Observer를 사용하여 구현했습니다. 스크롤이 최하단으로 갔을 때를 구독하다가 데이터를 조금씩 추가해 로딩시키는 방식이었습니다. 결과적으로 3초였던 로딩 시간은 최대 0.5초까지 줄일 수 있었습니다.

이슈 트래커 서비스 – 2020.06 ~ 2020.07

Stack – react, webpack, babel, react-router, react-redux, redux-thunk, styled-component, material ui

프로젝트에서 웹팩을 사용하기 때문에 production 모드일 때에는 빌드 최적화 작업을 해야 웹페이지가 빠르게 동작합니다. 그래서 React의 lazy 기능을 통해 code splitting을 진행하였습니다. 빌드했을 때 하나의 파일로 있었던 번들 파일이 lazy를 통해 동적 로딩한 개수만큼 나누어져 있는 것을 확인할 수 있었습니다. 파일 크기가 작아지니 빠르게 로딩하는 것도 개발자 도구의 Network 탭을 통해 확인했습니다.

## Self-introduction

프로젝트를 진행하면서 느낀 점이 가장 많은 부분이 "협업"입니다.

페어 프로그래밍에서의 협업은 드라이버와 내비게이터의 역할을 번갈아 가면서 진행됩니다. 페어를 진행하면서 가장 어려웠던 부분은 리드하는 것이었습니다.

두 명이 때문에 내비게이터일 때 코드를 진행하는 것이 필요합니다. 처음에 할 때는 역할을 분명하게 하지 않아 드라이버가 코드를 진행까지 했습니다. 그렇게 되니 소통이 적은 페어 프로그래밍이 되고, 결과적으로 실패했습니다. 문제 상황에서도 아이디어가 있는 사람이 코드를 짜니 피로도가 높았던 것입니다. 처음의 실패를 바탕으로 의도적으로 페어와 함께 회고하는 시간을 가졌습니다. 스몰 토크와 함께 지속할 점, 개선할 점에 대해 회고를 하니 얘기를 편하게 할 수 있는 분위기가 되고 페어 방식도 점점 개선되었습니다.

페어 규칙을 자세하게 적고, 고민한 것을 얘기하면서 의견을 맞추니 감정 소모도 전보다 감정 소모도 적었습니다.

하루 또는 일주일마다 회고함으로써 당장 프로젝트를 빨리하는 것보다 뒤를 돌아보며 반성해 나가게 되는 것으로 생각합니다. 그리고 이것이 지속하면 더 나은 결과물을 산출할 수 있을 것입니다.