

신성우

Frontend Engineer (신입)

[Github](#) · [TechBlog](#)

24siefil@gmail.com · 010-9147-1041



저는 . . . 엔지니어 입니다.

1. 더 나은 UX와 더 나은 코드에 대해 이야기하는 것을 좋아하는
2. 쉽게 만족하지 않기에 회의 때 말을 가장 많이 하는
3. 도전적인 미션과 밀도 높은 성장을 갈망하는

기술 스택

React, TypeScript, React-Query, Redux, storybook

기술 역량

- [React](#) 기반 SPA 개발에 능숙합니다. Headless 패턴과 Compound Components를 통해 [재사용성](#)이 높은 컴포넌트를 설계할 수 있습니다.
- [TypeScript](#)의 Utility Type과 Generics를 활용하여 효율적으로 JavaScript 정적 타입 분석을 할 수 있습니다.
- [모바일 앱 내 웹앱](#) 제작 경험이 있습니다. 해당 웹앱의 모든 UI는 반응형 디자인을 지원하며, App-like한 UX를 제공합니다.
- [React-Query](#)를 도입하여 [Redux](#) 기반 비동기 통신 [상태관리](#) 체계를 [이관/개선](#)한 경험이 있습니다.
- [storybook](#)을 통해 컴포넌트와 디자인 시스템을 테스트/관리 할 수 있습니다.
- [Git](#) 기반 분산 버전 관리 시스템 이용에 능숙합니다.

협업 역량

- 팀내 [애자일 방법론](#) 도입을 주도한 경험이 있습니다. 칸반 보드 기반 이슈 관리, 스프린트 단위 개발, 프리징 규칙, 스크럼 회의를 정착시켰습니다.
- Github, Git-flow 기반 CI 개발 프로세스에 익숙합니다. 팀내 PR 및 코드리뷰 프로세스를 정립/정착시킨 경험이 있습니다.
- [Figma](#)를 통해 디자이너와 긴밀하게 협업할 수 있습니다.
- [동료학습](#)을 중요하게 생각합니다. IT 창업 학회 내 알고리즘 스터디를 조직하고 리더로서 운영하고 있습니다.

경험

뱅크즈 창업 (7개월, 2022.04 ~ 진행중)

- 어린이를 위한 핀테크 서비스
 - 프론트엔드 엔지니어 담당
-

교육

Ecole 42 (20개월, 2021.03 ~ 2022.10)

- 프로젝트 기반 동료학습을 통해 학부 수준의 컴퓨터과학 학습
- [\[깃허브\] 수행한 프로젝트 모음](#)
- [\[기술 블로그\] 학습한 컴퓨터과학 이론 정리](#)

홍익대학교 (졸업 유예, 2016.03 ~ 2023.02)

- 학부 과정의 주요 컴퓨터과학 과목 수료
- GPA 3.84 / 4.50
- [\[깃허브\] 수행한 프로젝트 모음](#)

신촌 연합 IT 창업 학회 CEOS 수료 (6개월, 2022.03 ~ 2022.08)

- 웹 프론트엔드 개발 기초 학습
- 뱅키즈 MVP 개발

어린이를 위한 핀테크 서비스, 뱅키즈

참여 기간	7개월 (2022.04 ~ 진행중)
참여 인원	9명
담당 직무	프론트엔드 엔지니어
깃허브	github.com/bankidz/bankidz-client
서비스 개발기	24siefil.oopy.io/bankidz
기술 스택	React, TypeScript, React-Query, Redux, storybook

- 서비스 개요**
- 아이들이 재밌게 금융을 경험하고, 이를 통해 올바른 금융습관을 형성하도록 돕는 어린이 전용 핀테크 서비스. 주요 기능은 저축 기반의 실전 금융 경험 제공.
 - 기획자, 디자이너, 백엔드 엔지니어와 긴밀히 협업하며 초기 기획부터 출시 및 운영까지 A ~ Z를 경험
 - 푸시알림을 제외한 모든 비즈니스 로직을 웹 기술로 구현

- 기술 기여**
- 문제: [Redux](#) (Thunk) 기반 비동기 통신에 대한 상태관리 시 서버 데이터를 위한 규격화 되지 않은 로직의 비대화, 장황한 Boilerplate 수반, 클라이언트/서버 데이터에 대한 RTK Slice 혼재로 인한 응집도 악화
 - 해결: [React-Query](#)를 도입하여 [상태관리 체계 이관/개선](#)
 - 기록: [Redux는 본연의 역할에 충실하고 있는가?](#)
-
- 문제: 페이지 일부 영역의 데이터 fetch 실패에 대한 ‘재시도 UI’ 를 명령(절차)형으로 구현 시 안티패턴(Presentational-Container) 및 Props Drilling 발생
 - 해결: [선언형 컴포넌트](#)를 도입하여 관심사 분리
 - 기록: [지속 가능한 컴포넌트](#)
-
- 기여: 사용성 개선 및 App-like한 UX 제공
 - 구현: [React-Query](#) 기반 API Caching, Skeleton UI, react-transition-group 기반 Routing Animation, localStorage Caching 적용
 - 기록: [프론트엔드 엔지니어가 생각하는 UX](#)

기술 기여

- 문제: try/catch문 기반 분산된 API 에러처리 시 유지보수성 악화, 에러처리 누락 등 휴먼에러 발생
 - 해결: '에러 핸들러의 정의 위치, HTTP Status, 서비스 표준 에러 Code' 기준 우선순위에 따른 [전역 API 에러처리](#) 체계 설계/적용
-
- 기여: OAuth 전략, JWT 운용 체계 전담 개발
 - 회고: 초기에 [httpOnly & secure cookie](#)를 통한 JWT 관리로 높은 보안성 달성. 추후 웹뷰(EXPO) 관련 호환성 문제가 발생하여 JWT 관리를 [localStorage](#)로 이관하며, [Over Engineering](#)에 대해 배움.
-
- 문제: 모달 제어를 위한 Props Drilling 발생, 모달 액션 후 모달 닫기 등 부수적인 코드 반복 발생, 모달 관리의 파편화로 인한 유지보수성 악화
 - 해결: 모달 관리에 [전역상태](#)를 도입하여 컴포넌트간 커플링 해소, 모달 액션 후 부수적인 반복 로직 추상화, 중앙화(객체화)된 모달 관리를 통해 유지보수성 개선
-
- 기여: [storybook](#) 기반 컴포넌트 및 디자인 시스템의 테스트/관리 체계화. [Figma](#)의 디자인 시스템과 [ThemeProvider](#) 동기화를 통해 UI 개발 효율 제고.

협업 기여

- 기여: 팀내 [애자일 방법론](#) 도입 주도. 칸반 보드 기반 이슈 관리, 스프린트 단위 개발, 프리징 규칙, 스크럼 회의 정착을 통해 개발 프로세스 개선.
-
- 문제: 기존 디자인, 기획, API 수정이 발생하는 경우 일련의 대응작업이 순서에 따라 처리 되어야 하지만, 일부 작업이 누락된 채 파트간 이슈 이전
 - 해결: 칸반 보드에 이슈 템플릿(체크리스트, 이슈 이전 규칙)을 도입하여 [업무 프로세스 개선](#)
-
- 기여: Git-flow, Github Automated Kanban, PR/Issue Template을 도입하여 팀내 PR 및 코드리뷰 규칙 정립, CI 개발 프로세스 정착. 웹 프론트엔드 파트 코딩 컨벤션 정립.