HANCOM

REST API PLATFORM - 통합문서뷰어 통합 테스트 개선

공 희 진 / 솔루션개발팀

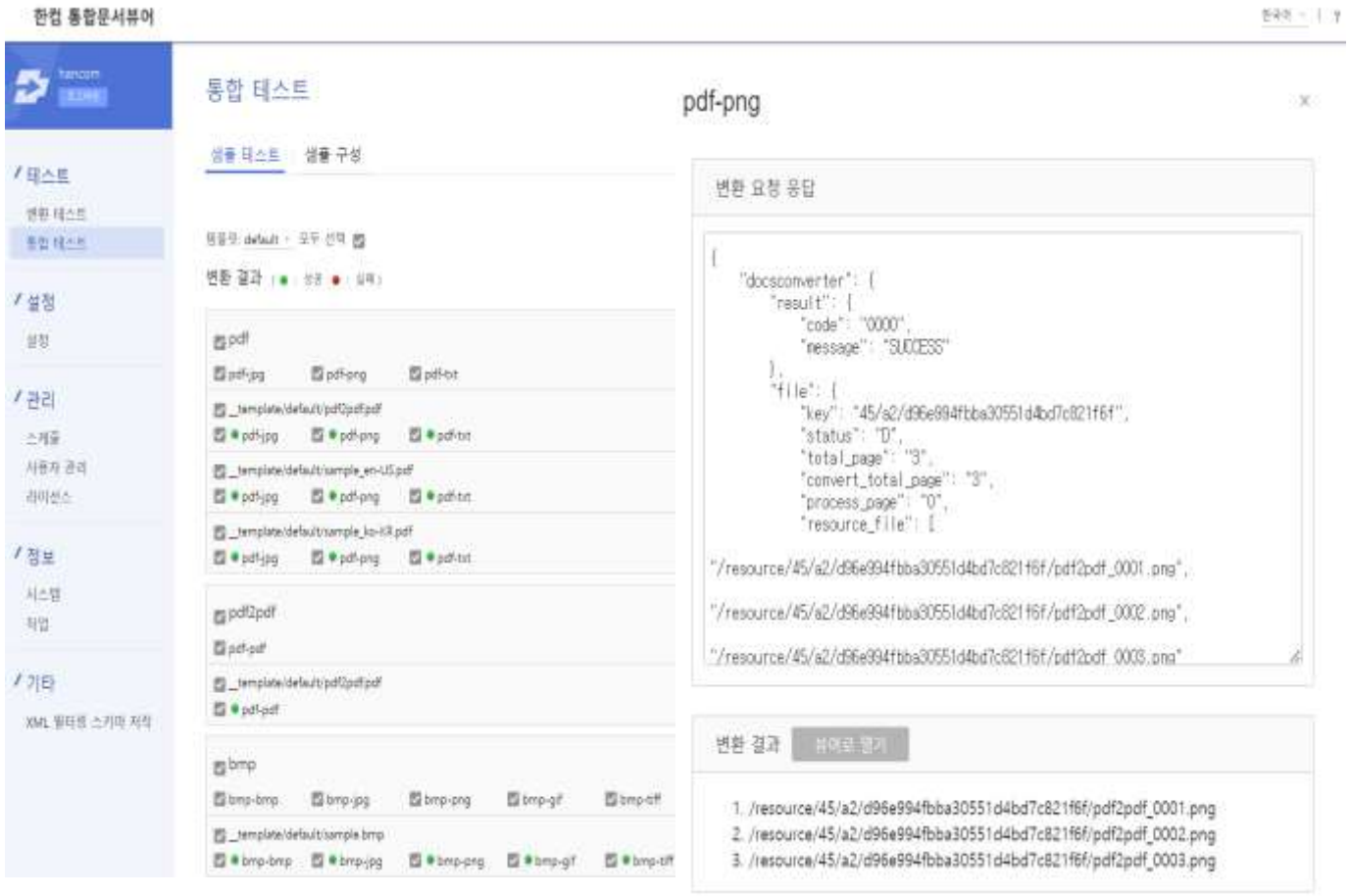
Contents

HANCOM

개요

- 통합 문서뷰어의 테스트 과정에서 발생한 문제 개선 -

기존 통합문서뷰어의 통합 테스트



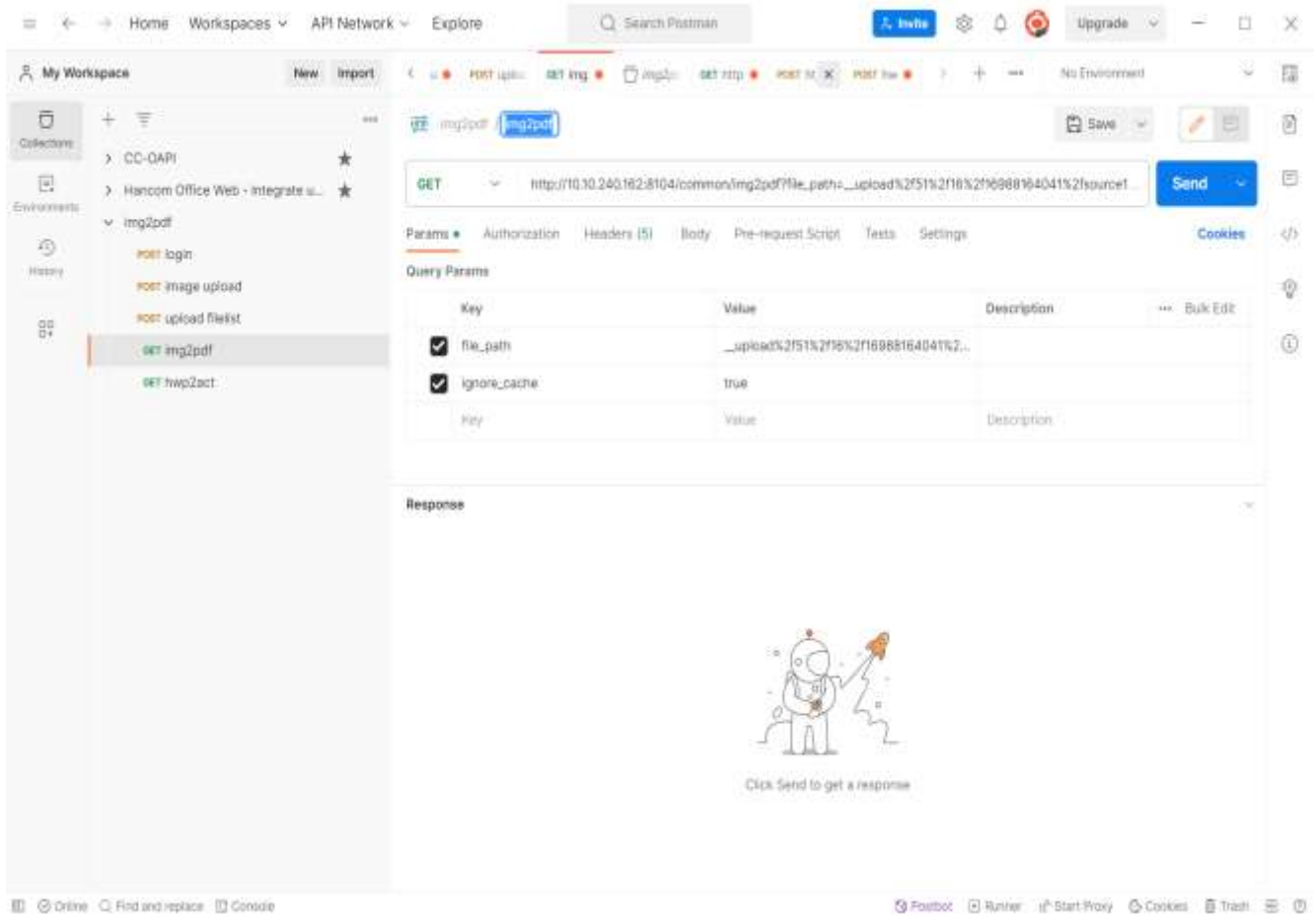
통합 문서뷰어 테스트 기능

- 문서 변환 결과 확인 가능

한계

- 각 포맷의 기본 변환 검사만 제공
- 응답 형식만으로 변환 성공 판단

불편하네.. 대체 하여 사용하는 테스트 툴은?



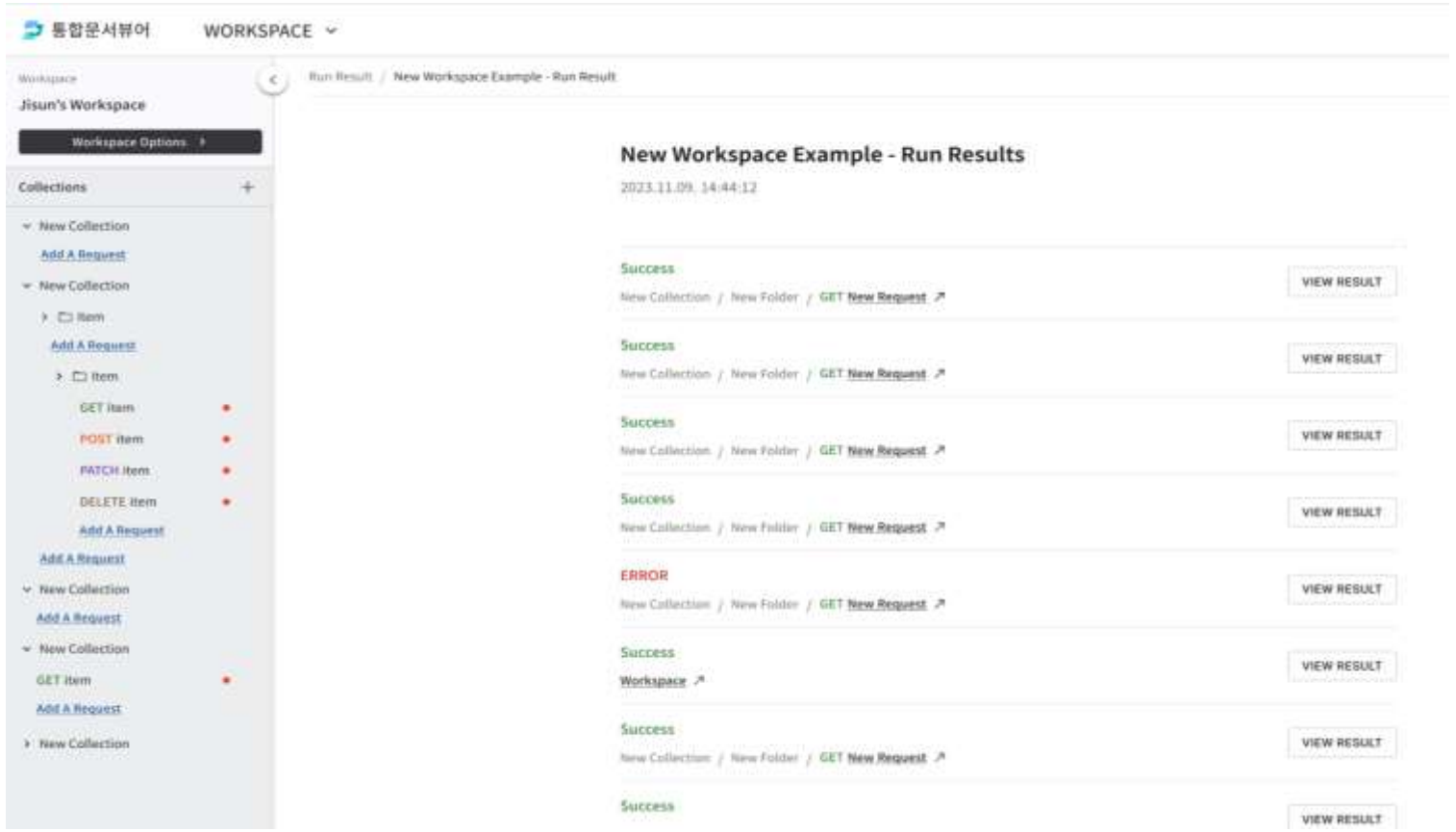
PostMan 효과적인 기능

- API 요청 문서화
- 컬렉션 관리 간편
- 요청/응답 UI가 직관적
- 요청 코드 생성의 편리함

단점

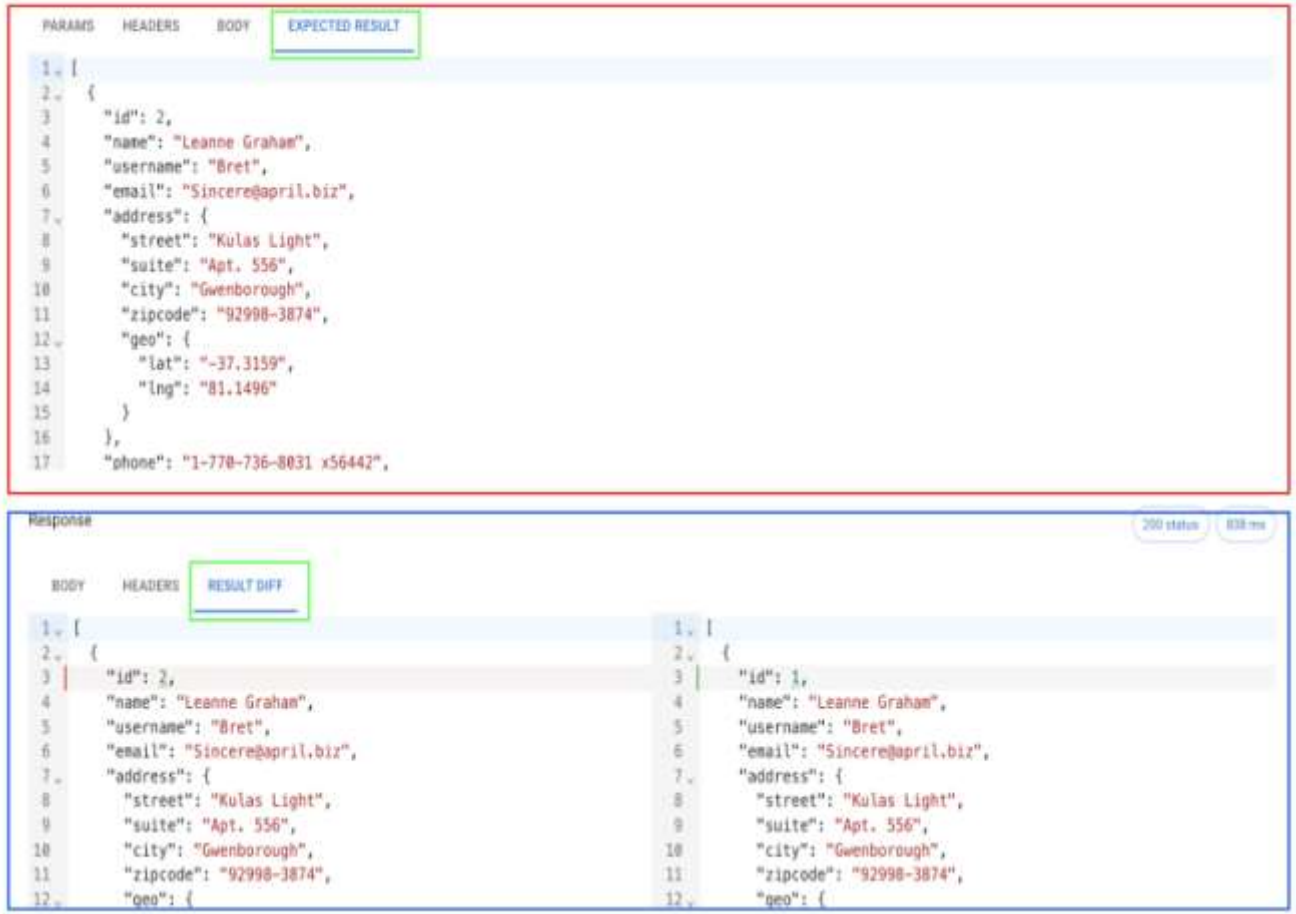
- 내부 사용 시, 유료 라이선스 필요

통합 테스트를 개선하자 !



- PostMan 의 장점 흡수
- 예상 결과와 변환 결과의 비교
- Run History
  - Run 실행한 기록 확인
- Run Workspace
  - 해당 Workspace의 하위 목록 전체 Run 가능

추가기능 - 예상 결과와 변환 결과의 비교

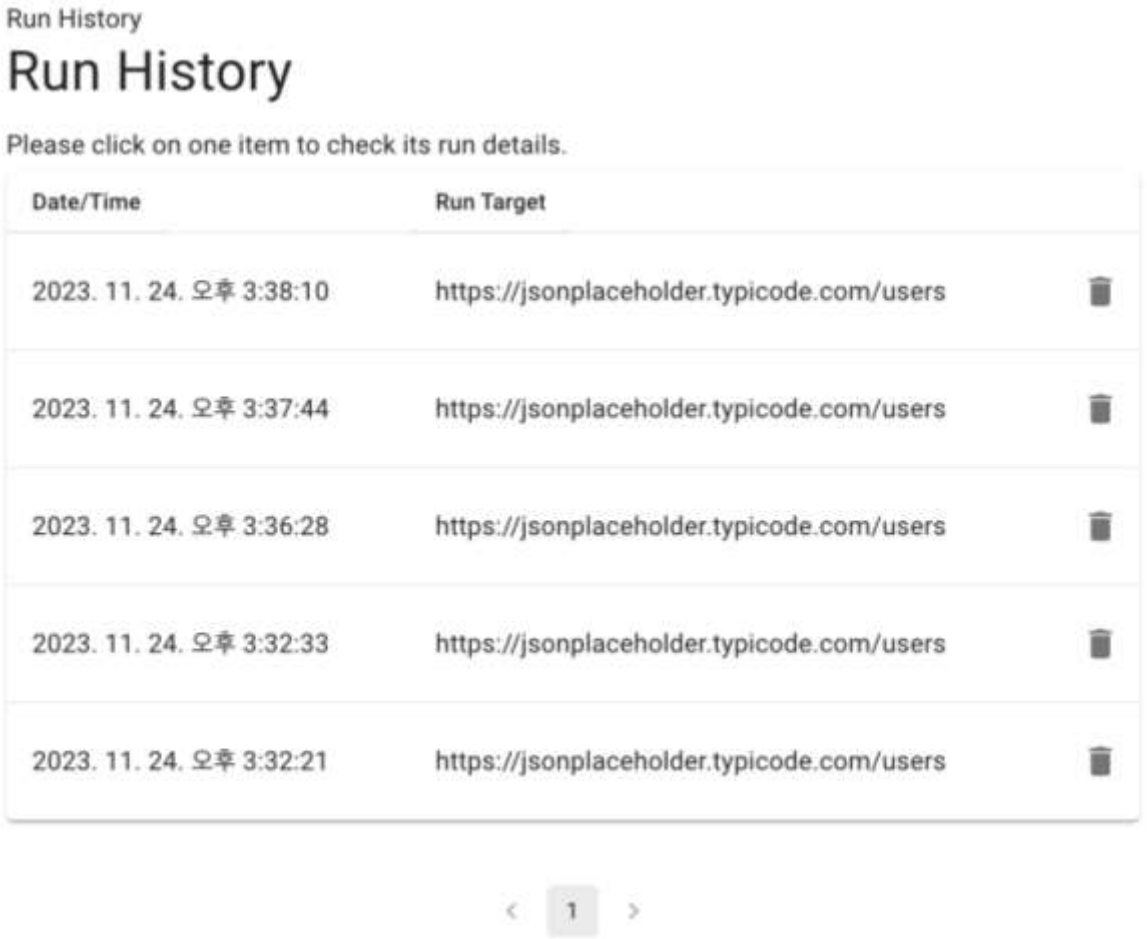


예상 결과 & 변환 결과 비교

- Request 탭에 예상 값 저장
- Response 탭에서 예상 값/ 응답값 비교 가능

추가기능 - Run 기록 확인

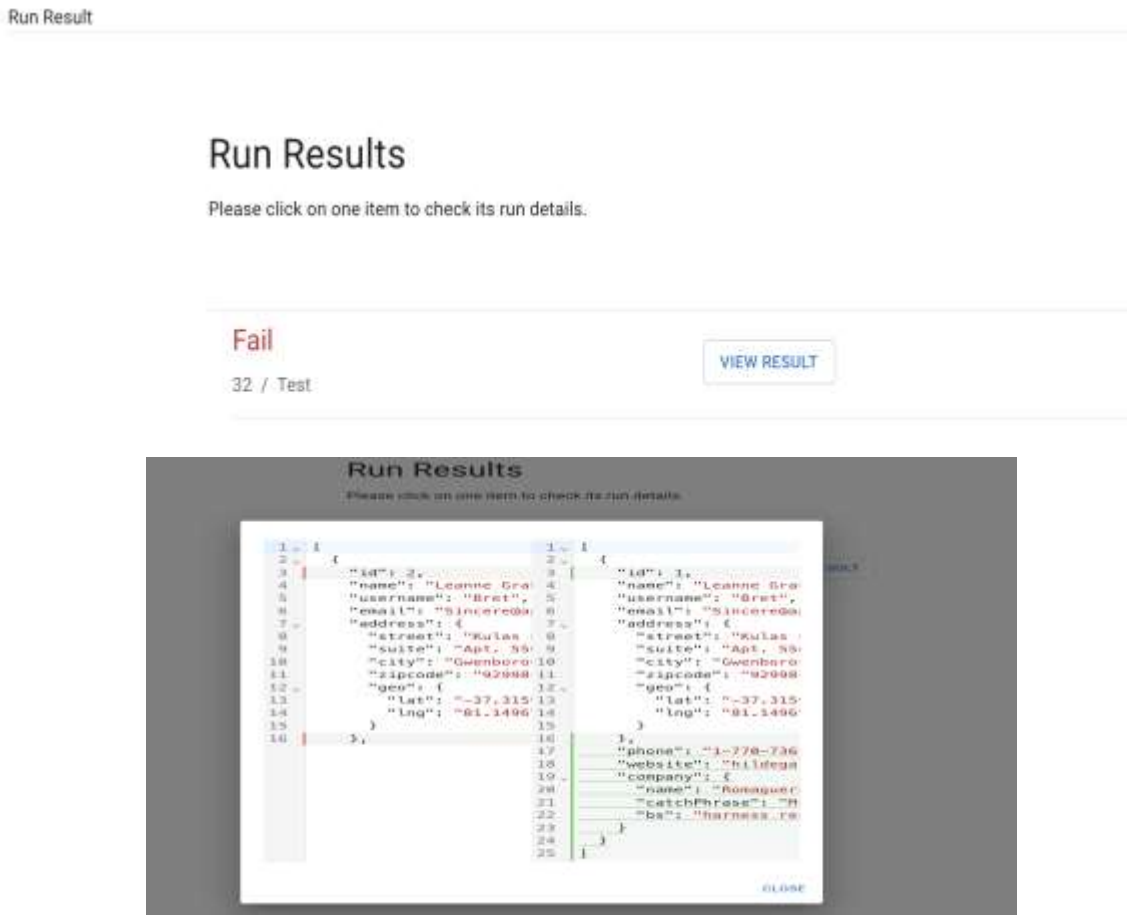
HANCOM



Run History 기록 확인

- Run 실행한 기록 확인
- 실행 결과에 대한 세부 정보 확인

추가기능 - Run 세부 정보 확인



Run Result 세부 정보

- 실행 결과의 성공 여부 확인
- 모달창으로 결과 값 확인

  - 예상 값 없을 경우

  - 예상 값 있을 경우,
    변환 성공 했더라도 예상 값과 다를 시, 실패

시연 링크

개발 일정

- 프로젝트의 주요 마일스톤 및 개발 일정-

Development History

REST API PLATFORM

2023. 8

프로젝트 기획 및 설계

관련 기술 스터디
초기 세팅 및 UI 설계
컴포넌트 별 UI 설계
아키텍쳐 설계

2023. 9

기본 기능 구현

Workspace 관리 기능 구현
Collection 및 Folder 추가, 기능 개발
Request 관리 기능 구현

2023. 10

기능 확장

HTTP 통신 및 기능 확장
저장 / 불러오기 기능 구현
(local storage 기반)
Tree 컨트롤 기능 구현
결과값 Diff 기능 구현

2023. 11

디자인 / 리펙토링

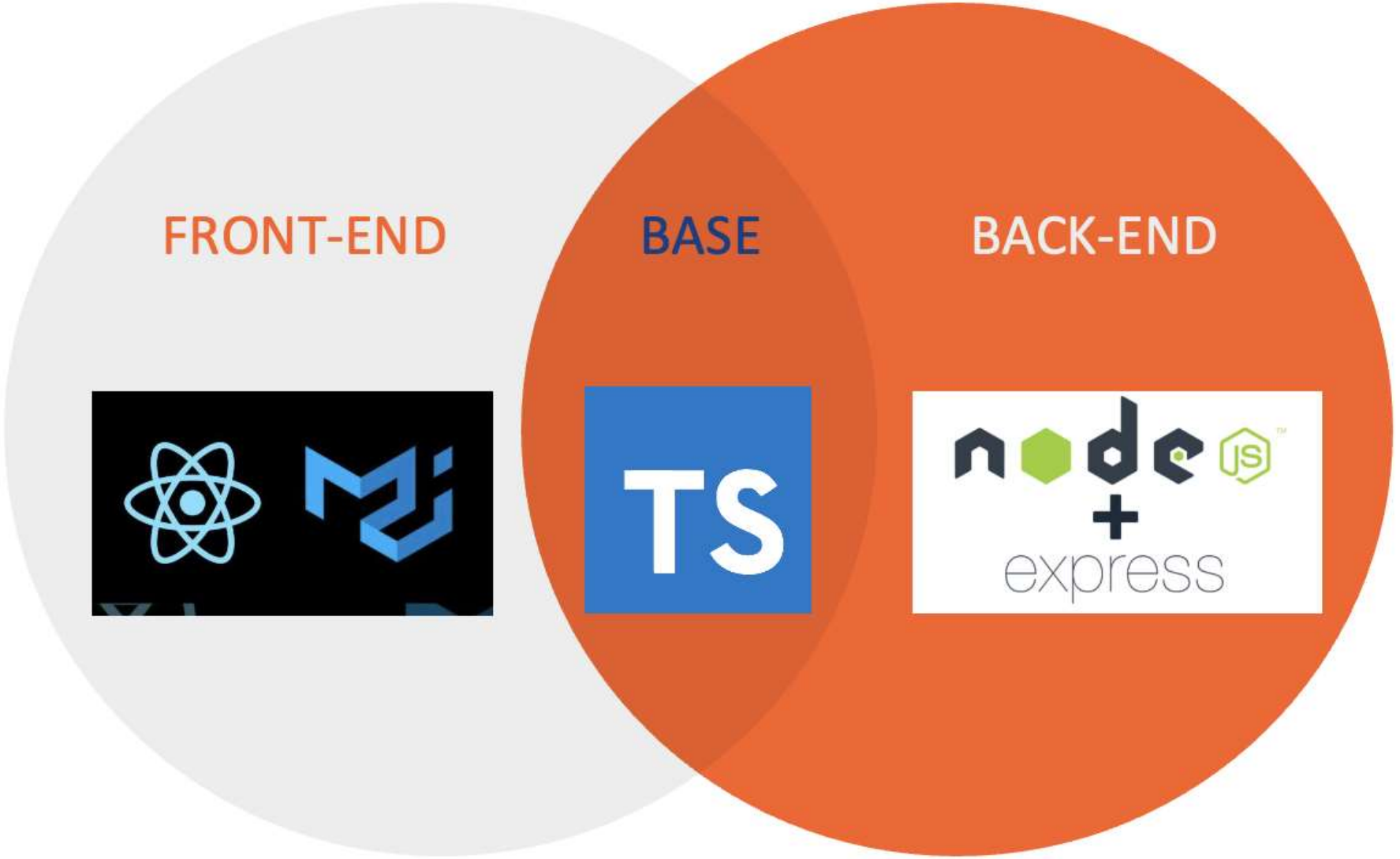
화면 디자인
구조 리펙토링
추가 기능 개발

스택 및 기술

- 사용된 기술 스택과 도구에 대한 간단한 소개-

스택 및 기술

HANCOM



개발 환경 구성

HANCOM

React-Router-Dom

Single Page Application 내부에서
Routing을 도와주는 라이브러리

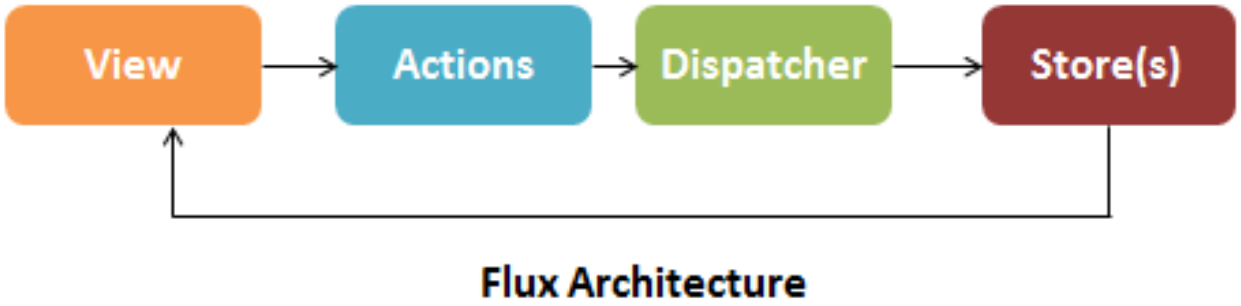
Redux

상태관리 위한 라이브러리
Flux / redux-thunk

Redux-Persist

Redux의 상태 유지를
위한 라이브러리

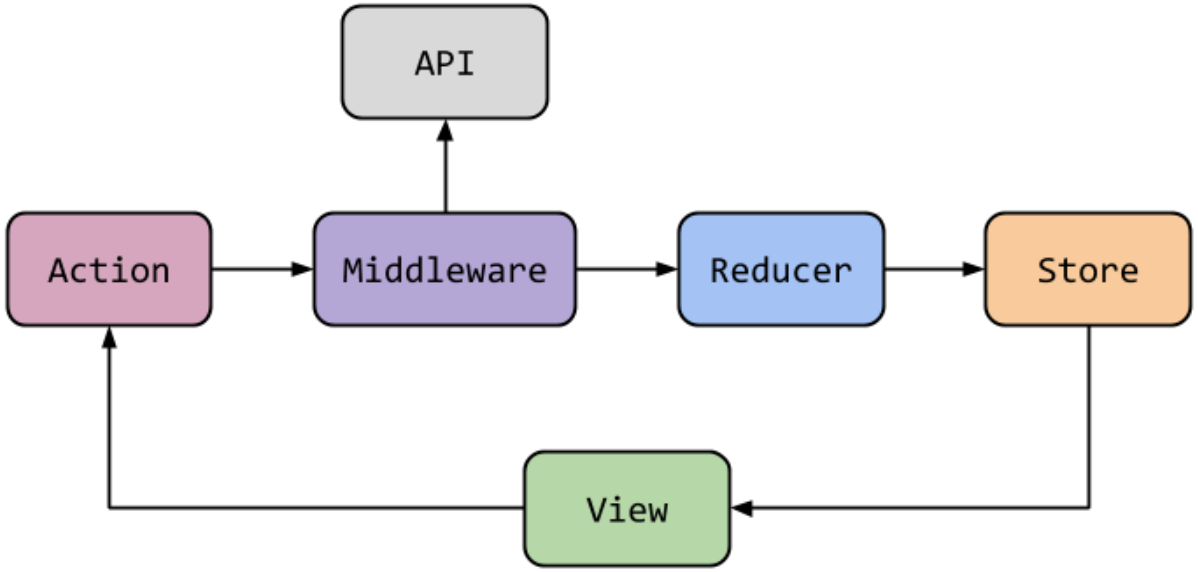
Redux - Flux



Flux Architecture

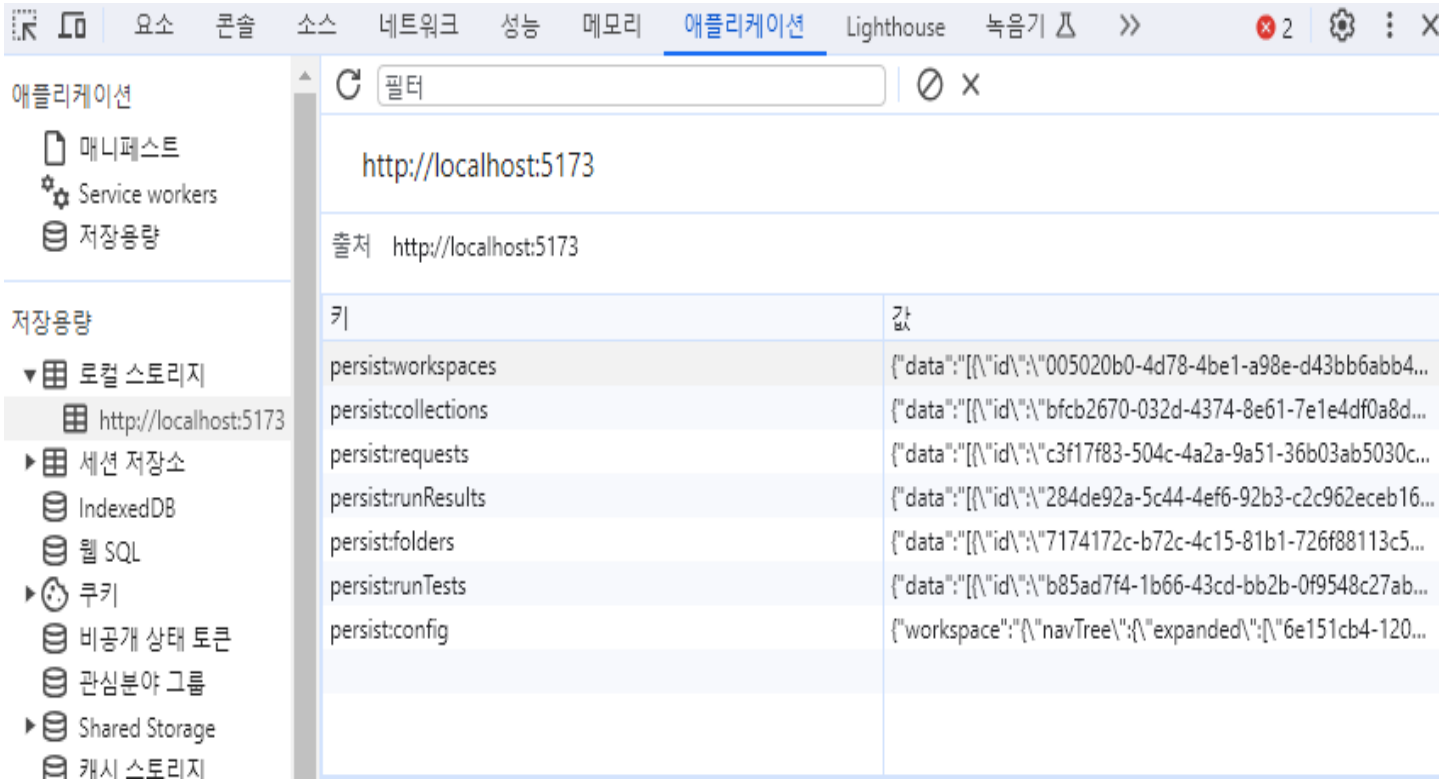
- 단반향 데이터 흐름
- Store의 데이터는 오직 Dispatcher 통해서 조작
- 데이터 수정 위치 추적 용이성

Redux-Thunk



- 비동기 작업 처리
- 매우 직관적이고 간단함
- 복잡한 데이터 수정

Redux-Persist

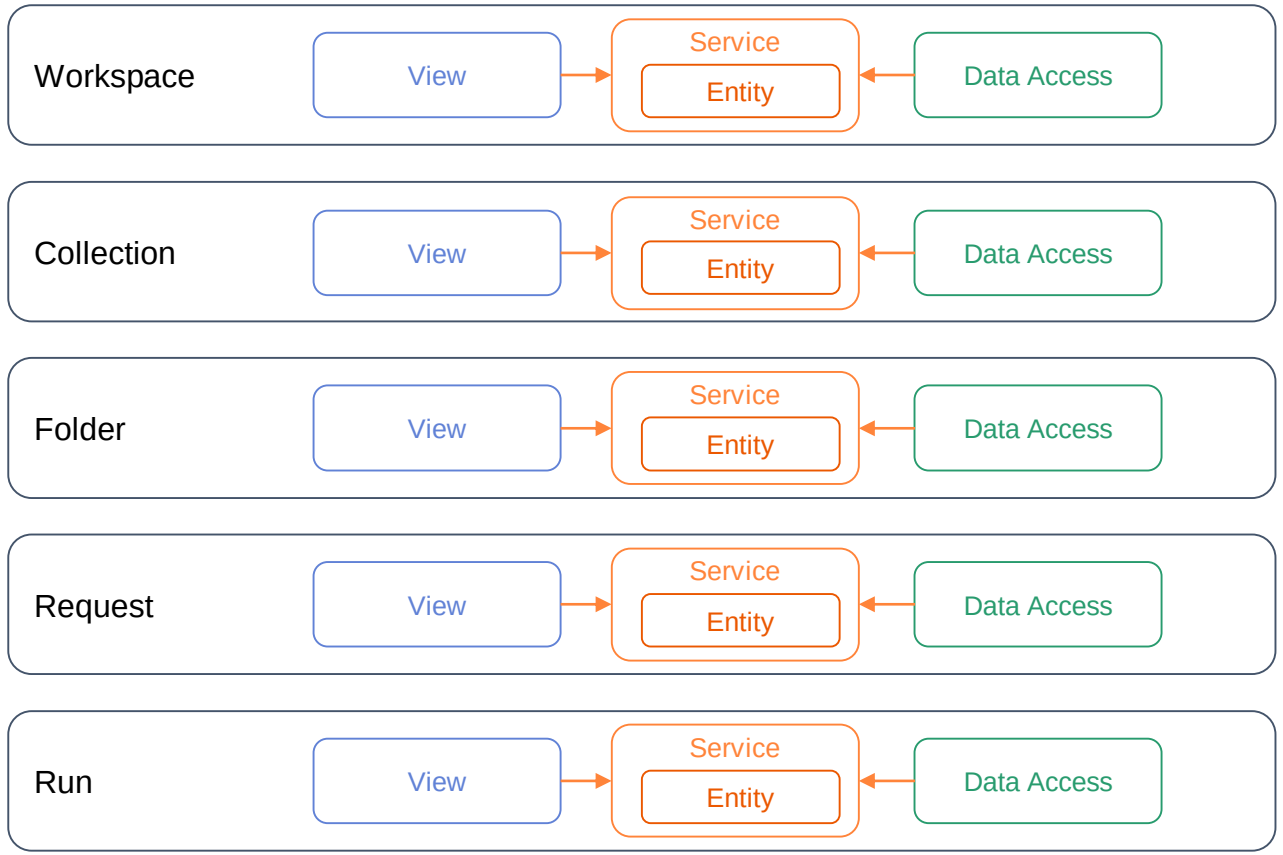


- state를 localStorage에 저장
- 새로고침시 state 유지 가능

설계

- EventStorming 원리를 적용한 프로젝트 설계 -

기능 지향 구조 (Feature Orientied Architecture)



- 핵심 기능 중심 개발
- 각 기능 내부에서 계층 구조 적용
- 의존성이 핵심 Entity를 향하도록 설계
- Event Storming을 적용하여 핵심 기능을 시각화

설계 – 이벤트 스토밍

HANCOM

"실제 기능과 데이터 간의 상호작용을 시각적으로 이해하고 공유하는 과정"

Domain Event

- 도메인에서 실제로 발생하는 **"결과"**

Command

- 도메인에서 특정 주체가 요청하는 **"행위"**

Actor

- Command를 발생시키는 **"주체"**

Aggregate

- Domain Event와 Command에 의해 처리되는 **"데이터"**

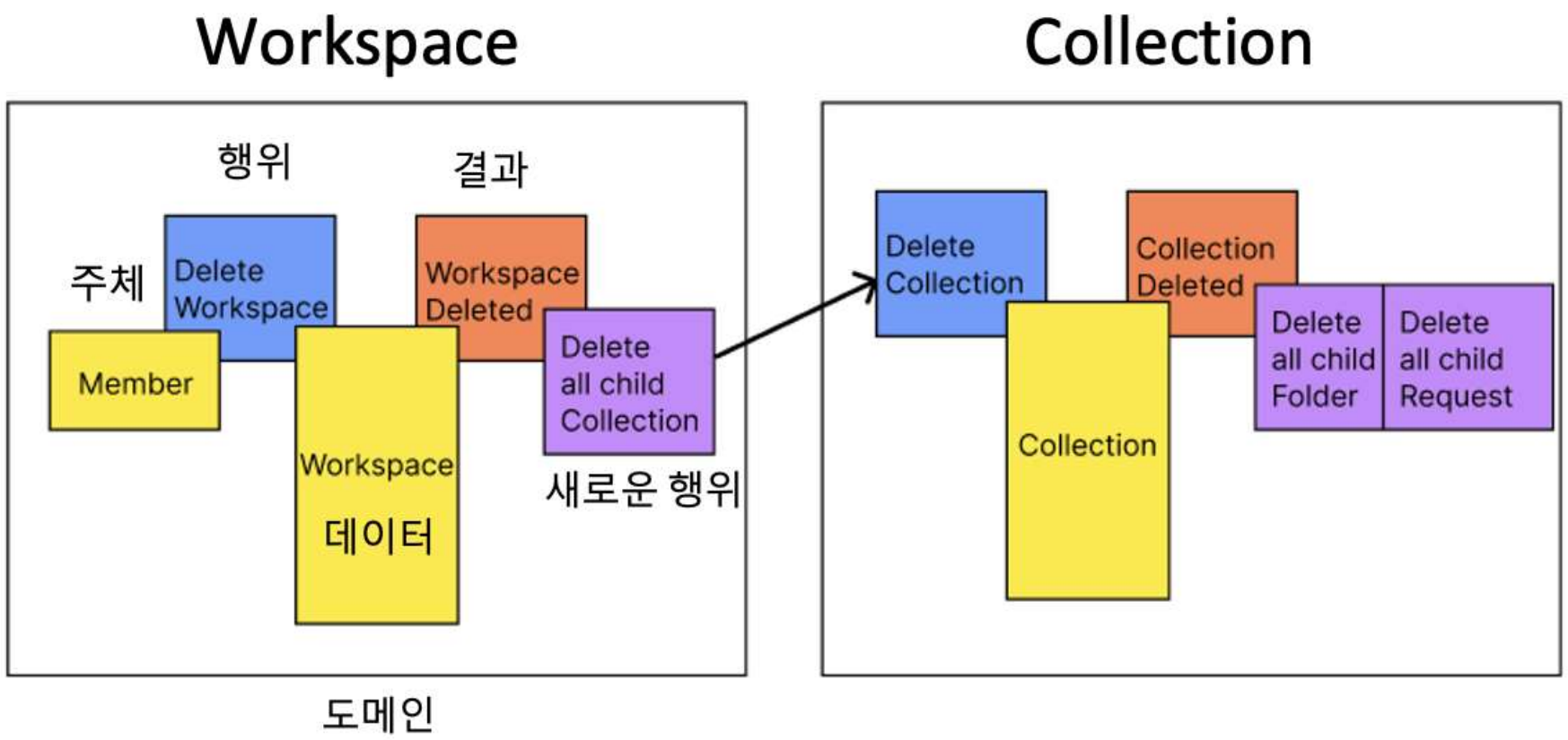
Policy

- 다른 Bounded Context에 영향을 주는 **"이벤트"**

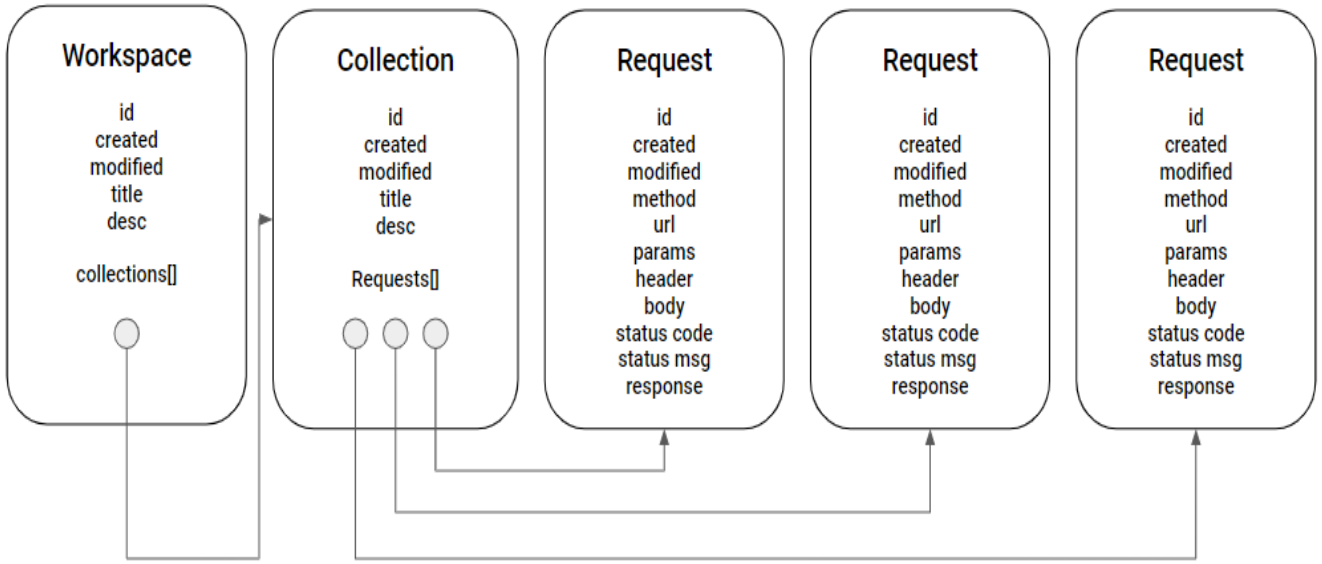
설계 - 서비스 분석

HANCOM

가정, Workspace를 삭제한다.

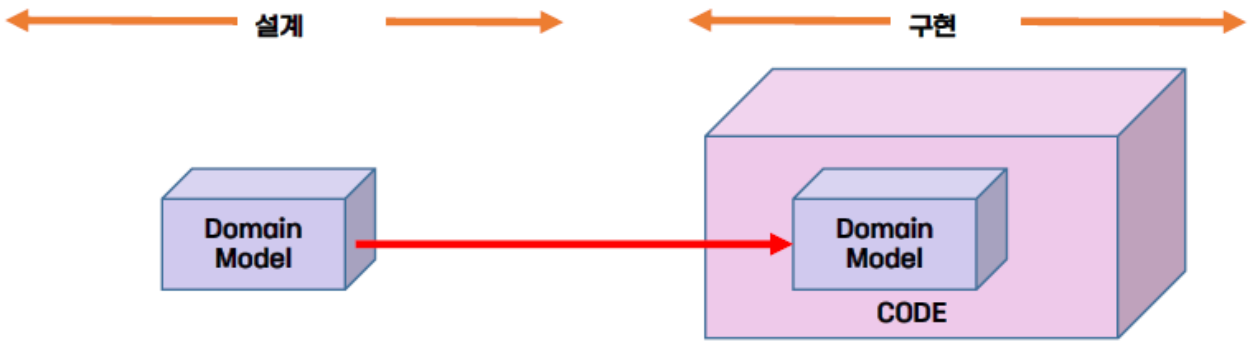


설계 - 데이터 정규화



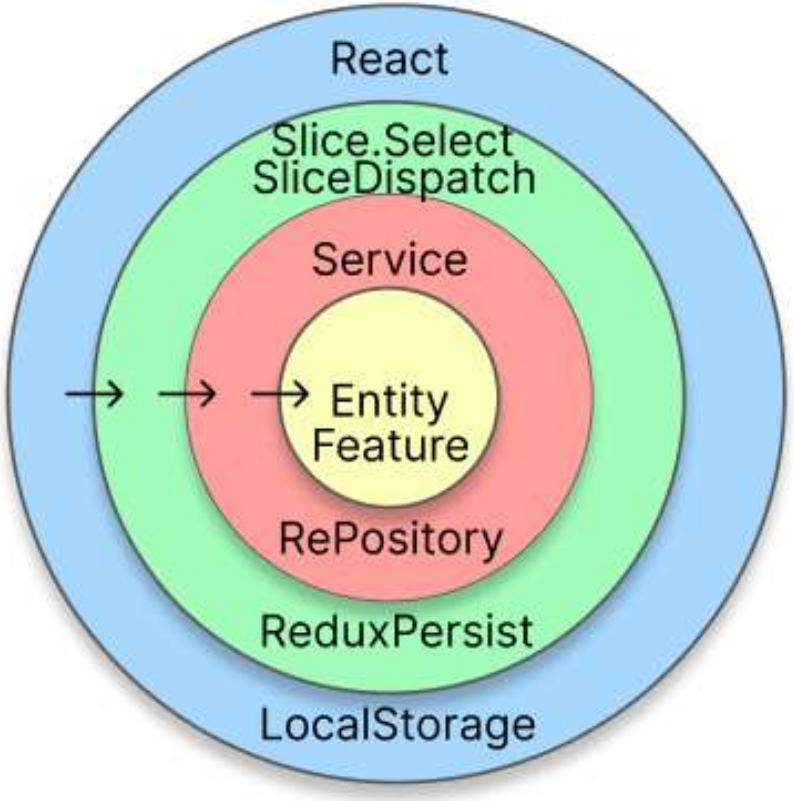
- 중복 최소화
- 효율적인 데이터 저장 프로세스
- uuid를 통한 검색 및 관리 용이성 강화

구현



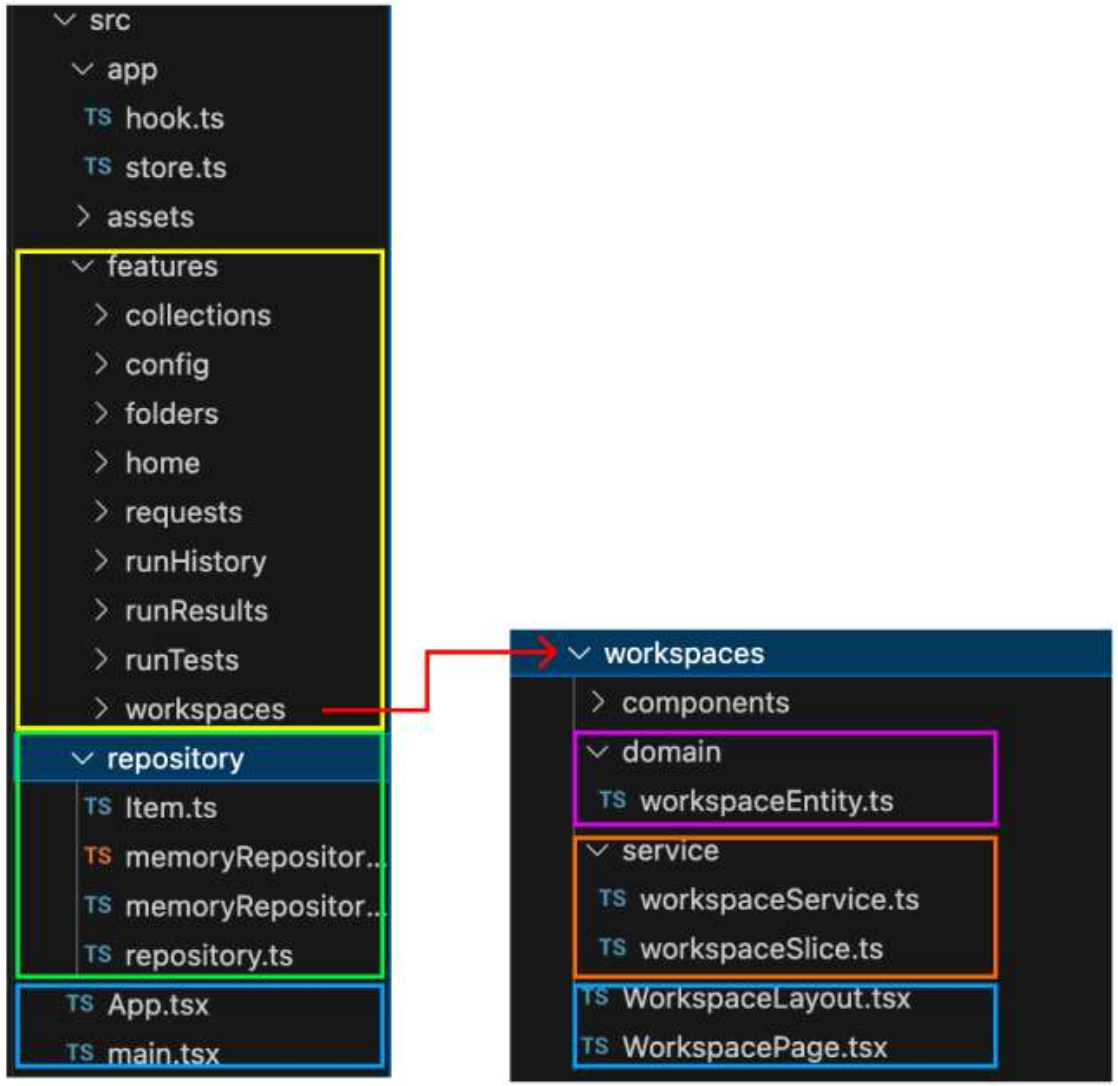
- 이벤트 스토밍을 통해 도메인 모델이 탄생
- 추상화된 도메인 모델 불변상태 로 고정
- 구현으로 도메인을 감쌈
- Clean Architecture 차용

프로젝트 아키텍쳐 - Clean Architecture



- Entity & Feature : 주요 기능 담당
- Service & Repository : 특정 기능 수행/데이터 저장소와 상호 작용
- Slice& ReduxPersist : 상태 관리 및 비동기 작업 처리
- React 및 LocalStorage : 사용자 인터페이스 및 데이터 효과적으로 관리
- 구현 변경시, 도메인 모델은 그대로 유지

디렉터리 구조



- hook.ts : 커스텀 hook
- store.ts : 상태 저장소
- features : 기능 로직
- repository : 저장소 로직
- domain : 기능을 추상화
- Service : 도메인 구현한 로직

아키텍쳐 링크

https://www.figma.com/file/HoEF33bdtHL2a7pPMRXCLZ/Rest-API-Platform-Architecture?type=whiteboard&node-id=0-1

향후 계획

2023. 12 중

디자인 퍼블리싱

디자인 적용

2023. 12 초

미구현 기능 개발

Run 기능
(Run Workspace,Run
Collecrion, Run Folder)

2023. 12 말

**PostMan 스펙으로
저장 기능 구현**

PostMan 호환
저장 / 불러오기 기능 구현

화면 디자인 가이드 링크

https://www.figma.com/file/FfPpXexxENLsTMnhSs847Q/RestAPI-Platform-PRJ?type=design&node-id=454-81784&mode=design

질문

감사합니다 ☺