

용어 및 개념

- 웰컴 페이지: 로그인 성공 시 보이는 페이지
 - 워크스페이스 리스트
 - 워크스페이스 생성, 선택
- 메인 페이지: 워크스페이스 선택 후 보이는 페이지
 - 워크스페이스: 팀원들이 모이는 공간. 팀원에게 DM 보내기 가능.
 - 채널: 워크스페이스 내의 그룹
 - 채팅: 같은 채널 내에서 채팅
 - 노트: 채널에 종속되어 있는 문서. 공동 편집을 위한 기능을 제공
 - 선점: 같은 노트에 접속해 있는 사람들 중 오직 한 명만 편집 가능

로그인

Email과 비밀번호를 입력하여 로그인

- 엔트리 포인트 URL: https://github.com/njsh4261/SGS_Last_Punch/blob/dev/src/frontend/web/src/components/Signin/index.tsx
- signinAPI URL: https://github.com/njsh4261/SGS_Last_Punch/blob/dev/src/frontend/web/src/Api/signin.ts

회원가입

회원가입 단계에 따라 조건부 렌더링

- 엔트리 포인트 : 단계에 따라 렌더링 및 API 사용

https://github.com/njsh4261/SGS_Last_Punch/blob/dev/src/frontend/web/src/components/Signup/index.tsx

- Signup API: 이메일 중복 검사, 인증코드 요청, 인증코드 검증, 회원가입

https://github.com/njsh4261/SGS_Last_Punch/blob/dev/src/frontend/web/src/Api/signup.ts

- apiHandler: API의 공통된 부분을 처리하기 위해 만든 handler

https://github.com/njsh4261/SGS_Last_Punch/blob/dev/src/frontend/web/src/Api/handler.ts

웰컴(워크스페이스 리스트 조회)

유저의 워크스페이스 리스트(wsList) 렌더링

- 엔트리 포인트: hook으로 얻은 wsList 렌더링

https://github.com/njsh4261/SGS_Last_Punch/blob/dev/src/frontend/web/src/components/Welcome/Body/WsList.tsx

- getWsListHook: wsList를 가져오는 API로 상태 업데이트

https://github.com/njsh4261/SGS_Last_Punch/blob/dev/src/frontend/web/src/hook/getWsList.ts#L8

- getWsListAPI: 주어진 페이지의 목록을 가져오는 API

https://github.com/njsh4261/SGS_Last_Punch/blob/dev/src/frontend/web/src/Api/workspace.ts#L8

워크스페이스 개설

워크스페이스 이름, 기본 채널 이름 입력 후 개설

- 엔트리 포인트: 워크스페이스/채널 이름, 개설 단계 렌더링

https://github.com/njsh4261/SGS_Last_Punch/blob/dev/src/frontend/web/src/pages/CreateWs.tsx

- createWsHook: 상태 관리 및 API 호출

https://github.com/njsh4261/SGS_Last_Punch/blob/dev/src/frontend/web/src/hook/createWs.ts

- createWsAPI: 워크스페이스 개설 API

https://github.com/njsh4261/SGS_Last_Punch/blob/dev/src/frontend/web/src/Api/workspace.ts#L48

메인

내 유저 정보 가져오기, 현재 채널 store에 업데이트, 알림 해제

현재 채널: url params의 channelId

- [엔트리 포인트](#): 각각의 hook 호출, 사이드 바 상태 관리(숨기기, 나타내기), 알림 해제 로직
- Hooks: [getSelfInfoHook](#), [updateChannelStoreHook](#)
- Dispatch: redux store를 업데이트 시킴. Channel의 경우 미들웨어인 redux-saga가 API를 호출하면서 응답 전에는 loading state가 되도록 함. ([user](#), [channel](#), [channelList](#), [userList](#))
- API: [getSelfInfoAPI](#), [getChannelInfoAPI](#)

메인/사이드 바

워크스페이스 채널/DM 목록 렌더링, 선택한 채널 id 값으로 URL 변경, 새 채팅이 있을 경우 알림(css효과로 하이라이트). 채널 선택 시 해당 채널에 종속된 노트 리스트 렌더링

- [엔트리 포인트](#): 워크스페이스의 채널 및 멤버 목록 조회와 채널 선택 hook 호출, store 값 이용
- Hook: [getChannelsAndMembersHook](#), [selectChannelHook](#)
- API: [getMembersAPI](#), [getChannelsAPI](#)

워크스페이스의 정보들을 상위 컴포넌트가 아니라 여기서 가져 온 이유는 content 부분인 Chat과 Note에서는 직접적으로 필요한 정보가 아니기 때문에, props drilling을 피하기 위해서 입니다.

메인/채팅

선택된 채널에 대한 채팅을 보여줌. 채팅 메시지 상태 관리, 소켓 관리, 스크롤 관리를 합니다.

- [엔트리 포인트](#): 각각의 hook 호출과, userDictionary로 유저 id와 name을 맵핑하여 렌더링
- Hook: [chatHook](#)(메시지 입력 핸들링, 최신 메시지 요청), [chatSocketHook](#)(연결 및 pub/sub), [chatScrollHook](#)(새 메시지 입력 시 아래로 스크롤, 위로 스크롤 시 예전 메시지 요청)
- API: https://github.com/njsh4261/SGS_Last_Punch/blob/e7c58522c9aed8e86473104cb092fecfec51b7f2/src/frontend/web/src/Api/chat.ts

chatSocketHook에서 구독의 경우, 새 메시지가 왔을 때의 알림을 위해 모든 채널, 모든 유저에 대해 구독을 함

메인/노트(1)

채널 하위에 있는 노트 선택 시 채팅 화면에서 노트 화면으로 전환. 타이핑을 먼저 치는 사람이 선점권을 획득하고 편집을 할 수 있음. 나머지는 편집 내용을 실시간으로 업데이트하여 볼 수만 있음.

- 주요 변수 및 함수

- Owner: 선점자. 현재 노트에서 유일하게 편집할 수 있는 선점권을 지닌 유저
- Op: operation. 선점자가 일으킨 편집 액션. opQueue에 모아서 주기적으로 업데이트
- lock/unlockNote: lock은 선점권 요청(소켓 서버에서 먼저 온 요청에 선점권을 줌)
- updateNoteAll: 노트의 모든 내용을 REST API로 저장 요청.
- editor: slate editor

메인/노트(2)

- [엔트리 포인트](#): 각각의 hook을 호출하고 typing 및 update에 관한 handler가 정의되어 있습니다.
 - `keydownHandler`: 선택 및 선택권에 대한 핸들링
 - `changeHandler`: 선택자에게만 사용되는 핸들러
- Hook
 - [noteSocketHook](#): socket 연결과 구독 및 콜백함수, 이벤트에 따른 메시지를 보내는 함수들이 정의
 - [noteSetupHook](#): 노트의 현재 값을 요청하여 적용, 선택권을 가진 채로 브라우저 종료 시에 대한 예외처리, 특수키로 선택권 획득에 대한 예외처리
 - [noteOPintervalHook](#): `opQueue`가 있다면 주기적으로 `op update`([Dan Abramov의 useInterval](#)을 참고)
- API: https://github.com/njsh4261/SGS_Last_Punch/blob/dev/src/frontend/web/src/Api/note.ts

부족한 코드를 봐주셔서 감사합니다!