초기 셋팅

* 노드 LTS 버전 설치(<https://nodejs.org/ko/>)
* Yarn 설치 (npm install -g yarn 또는 <https://yarnpkg.com/lang/en/docs/install/#windows-stable> 에서 설치)
  + 프로젝트에서 노드 패키지 관리를 yarn 사용하므로
  + 폰트어썸 프로 적용
    - (참고: <https://fontawesome.com/how-to-use/on-the-web/setup/using-package-managers>)
* 프론트 프로젝트 root 에서 yarn install 실행
* Prettier 사용하기
* 깃
  + 깃 클론 (ssh://jhlee2@172.20.0.7:7722/repo/git/mobon\_admin\_v2\_frontend.git)
  + 브랜치
    - 개인 브랜치 만들어서 관리
    - 개인 브랜치에서 작업할 때 최신 소스에 개발 할 수 있도록 develop 풀 받은 후 작업
    - 개인 브랜치 작업 후 develop 에 적용
    - master 브랜치는 여러 개발자들이 적용해둔 최신 develop 소스를 당겨옴

파일 구조

* App.js
  + Root 인 index.js 에서 App.js 호출 > App.js에서 페이지 라우팅
  + 2depth 까지만 라우팅 / tab 라우팅은 해당 페이지 component 내에서 이루어짐
  + 권한 관리에서 페이지 추가 및 url 변경한 경우 App.js에서 추가 및 변경된 url 에 맞는 page component 맵핑 해줘야함
* pages
  + 각 페이지별로 폴더 구분
  + templates 폴더는 페이지 별로 공통으로 가져다 쓰는 템플릿 컴포넌트 존재함
* modules
  + 리덕스 관련 모듈들
* lib
  + api폴더
  + 각종 공통 함수 등 존재
* styles
  + globals.scss 에 date picker를 제외한 사용하지 않는 css 정리 필요 (js in css 사용하고 있으므로)
* images
  + svg 및 각종 필요한 이미지파일들
* constants
  + 정적인 변수 등 선언되어 있는 파일들
* components
  + 아토믹패턴으로 정리
  + atoms
  + molecules
  + organisms

다국어 처리

* i18next 리액트용 사용
* 한국어를 키값으로 하여 en.json 파일에 해당 영문 번역 값 json 정의해야함

공통 컴포넌트들은 components 폴더안에 아토믹 패턴으로 파일 생성하고 페이지 작업시 pages 폴더에 원하는 페이지 폴더 및 파일 생성하고 App.js에 해당 url 에 페이지 컴포넌트 매핑하여 결과 확인하면 됩니다.