



컨벤션

Air bnb Style Guide

참고자료 - <https://github.com/tipsy/javascript-style-guide>

그라운드룰

- 폴더명은 전부 소문자에 복수형
 - components
 - pages
- 페이지랑 컴포넌트 이름은 PascalCase
 - MyPage.tsx
 - MainPage.tsx
- 변수랑 함수는 camelCase
 - const userHandle
- 함수는 화살표 함수, 익명함수 보다는 핸들로 해서 바깥으로 빼주세요

```
// 익명함수 사용하지 않고 함수 선언 후 핸들 예시
const handleClick = () => {
  console.log('버튼 클릭');
}
<button onClick={handleClick}>버튼</button>
```

- 상수를 명시 해준다. 대문자 스네이크 케이스로

```
enum HTTP_STATUS {
  BAD_REQUEST = 400,
  UNAUTHORIZED = 401,
  FORBIDDEN = 403,
  NOT_FOUND = 404,
  INTERNAL_SERVER_ERROR = 500,
}

const USER_STATUS = {
  ADMIN : 1,
  STUDENT : 2,
}

if (user.status === 1){
  console.log('이러지말고')
}
if (user.status === USER_STATUS.ADMIN) {
  console.log('이렇게 명시')
}
```

```
// HTTP 응답 상태 코드 상수 객체
enum HTTP_STATUS {
  BAD_REQUEST = 400,
  UNAUTHORIZED = 401,
  FORBIDDEN = 403,
  NOT_FOUND = 404,
  INTERNAL_SERVER_ERROR = 500,
}

const HTTP_STATUS = {
  const BAD_REQUEST = 400,
  const UNAUTHORIZED = 401,
  const FORBIDDEN = 403,
  const NOT_FOUND = 404,
  const INTERNAL_SERVER_ERROR = 500,
}

// 유저 상태 및 정보 유니온 타입으로 선언
type UserState =
  | { status: 'loading' }
  | { status: 'success', data: UserData }
  | { status: 'error', error: string };

type UserData = {
  id: number;
  name: string;
  email: string;
};

const getUserData = async () : Promise<UserState> => {
  try {
    const response = await axios.get(`대충 유저 정보 가져오는 api`);
    if (response.status === HTTP_STATUS.OK) {
      return updateUserState(response.status, response.data);
    }
    return updateUserState(response.status);
  } catch (error) {
    if (error.response) {
      return updateUserState(error.response.status, error.response.data);
    } else {
      return { status: 'error', error: '네트워크 에러가 발생했습니다.', code: HTTP_STATUS.INTERNAL_SERVER_ERROR };
    }
  }
}

```

- 컴포넌트나 페이지를 만들면 index.ts 안에 집어넣고 export 해준다

```
├─ index.html
├─ index.css
├─ .eslintrc.cjs
├─ .prettierrc
├─ package.json
├─ package-lock.json
├─ tsconfig.json
├─ src
│   ├── apis
│   ├── components
│   │   ├── index.tsx
│   │   └─
│   │       └─ tree
│   │           ├── pages.tsx
│   │           ├── render.tsx
│   │           └─ design.scss
├─ pages

```

```
| | | index.tsx
| | | loading.tsx
| | | 404.tsx
| | | _app.tsx
| | |
| | | | | index.tsx
| | | | | [id].tsx
| | | |
| | | | | index.tsx
| | | | | [id].tsx
| | | | user
| | | | | index.tsx
| | | | | [id].tsx
| | |
| | stores
| | types
| | main.tsx
| | routes.tsx
| | vite-env.d.tsx
```

- 테일 윈드 CSS 사용
 - 하나의 CSS 파일에 테일 윈드 import 해서 하나의 Class로 만들어서 사용 → 클린 코드!