

• XEXYMIX

XEXYMIX CLONE PAGE GUIDE

김범서

I N D E X

목 차

- 기획의도
- 사이트 구현
- 향후 개발 방향

기획 의도

개발 방향

- 데이터 활용
- 데이터 필터링
- 컴포넌트 재사용
- 웹, 모바일 웹 미디어쿼리
- 로그인 기능 구현 – REST API, axios, recoil

Font & Color

Montserrat

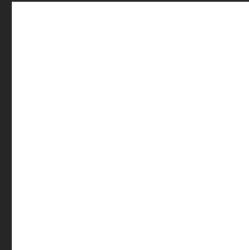
Noto Sans KR



#000



#ccc



#fff

상태관리 [Recoil] 전역상태관리 라이브러리



상태관리 [Recoil] 전역상태관리 라이브러리

▼ recoil
atoms.js

```
import { atom } from "recoil";

export const userInfoState = atom({
  key: "userInfoState",
  default: {
    accessToken: null,
    id: null,
    nickname: null,
    profileImage: null,
    thumbnailImage: null,
    isLogin: false,
  },
});
```

```
import { userInfoState } from "../recoil/atoms";
import { useRecoilValue, useRecoilState } from "recoil";

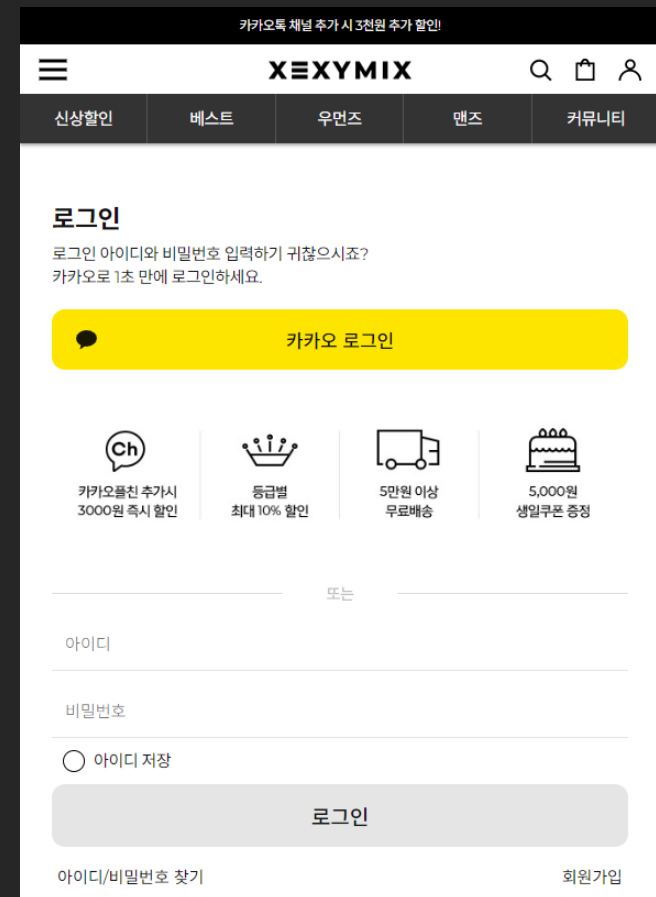
const userInfo = useRecoilValue(userInfoState);
const [userInfo, setUserInfo] = useRecoilState(userInfoState);
```

카카오 로그인 기능 구현

REST API 방식

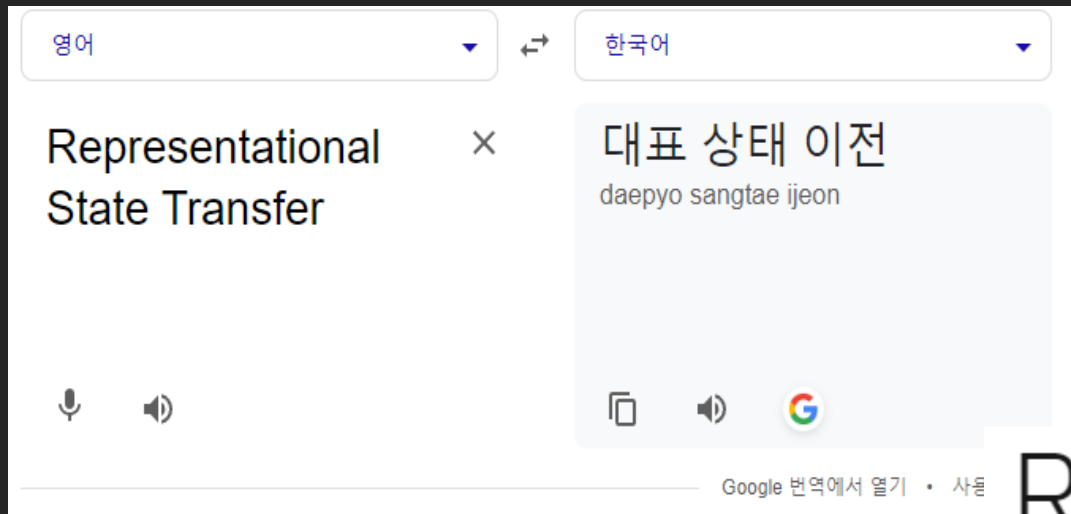
axios

async & await



REST API

GET, POST 와 같은 상태를 주고받으며 API와 통신하는 것

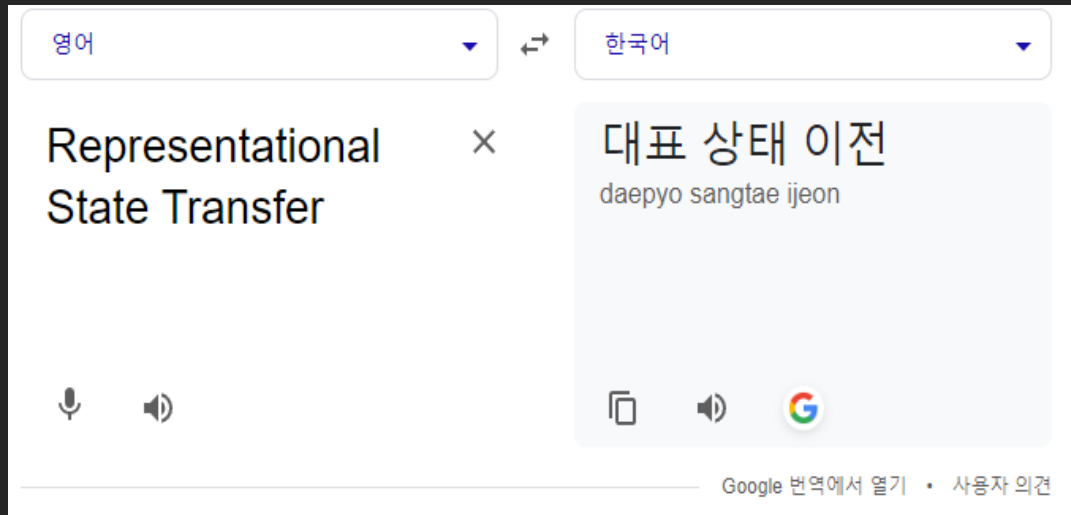


REST API의 작동 방식

- GET : 데이터 조회
- POST : 데이터 등록 및 전송
- PUT : 데이터 수정
- DELETE : 데이터 삭제

REST API는 HTTP 요청을 통해 통신함으로써 리소스 내에서 레코드(CRUD 라고도 함)의 작성, 읽기, 업데이트 및 삭제 등의 표준 데이터베이스 기능을 수행합니다. 예를 들어, REST API는 GET 요청을 사용하여 레코드를 검색하고, POST 요청을 사용하여 레코드를 작성하며, PUT 요청을 사용하여 레코드를 업데이트하고, DELETE 요청을 사용하여 레코드를 삭제합니다. 모든 HTTP 메소드는 API 호출

REST API



GET, POST 와 같은 상태를 주고받으며 API와 통신하는 것

- GET : 데이터 조회
- POST : 데이터 등록 및 전송
- PUT : 데이터 수정
- DELETE : 데이터 삭제

메서드	URL
POST	<code>https://kauth.kakao.com/oauth/token</code>

REST API

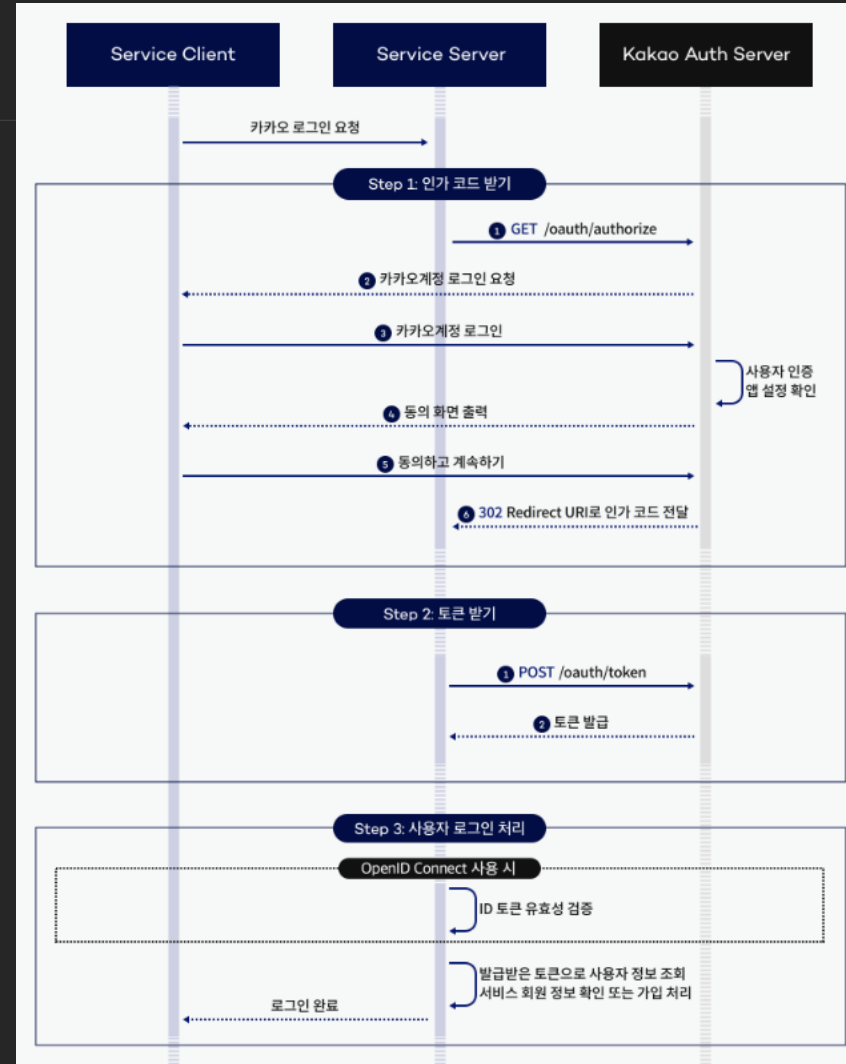
```
// Access Token 받아오기
// async 를 함수 앞에 선언해주면 비동기 함수가 됨.
// (promise 객체를 반환하지 않아도 자동으로 반환함)
const getAccessToken = async () => {
  if (accessTokenFetching) return;
  console.log("getAccessToken 호출");
  try {
    setAccessTokenFetching(true);
    // await는 promise의 상태가 바뀌면 실행됨(then 과 비슷함).
    const response = await axios({
      method: "POST",
      headers: {
        "Content-type": "application/x-www-form-urlencoded;charset=utf-8",
      },
      url: "https://kauth.kakao.com/oauth/token",
      data: makeFormData({
        grant_type: "authorization_code",
        client_id: REST_API_KEY,
        redirect_uri: REDIRECT_URI,
        code: KAKAO_CODE,
      }),
    }),
  );
  const accessToken = response.data.access_token;
  console.log(response);

  setAccessTokenFetching(false);
  getProfile(accessToken);
} catch (error) {
  console.error("Error:", error);
  setAccessTokenFetching(false);
}
};
```

GET, POST 와 같은 상태를 주고받으며 API와 통신하는 것

메서드	URL
POST	https://kauth.kakao.com/oauth/token

카카오 로그인 REST API



향후 개발 방향

기능 구현

페이지 로그인, 카카오 로그인 연동

로그아웃,
카카오와 함께 로그아웃 기능 구현

위시리스트 추가 기능 구현

서브 및 상세 페이지 구현

위시리스트 페이지 구현 예정

커뮤니티 게시판 구현 예정