|  |
| --- |
| import axios from 'axios';  const apiClient = axios.create({  baseURL: process.env.NEXT\_PUBLIC\_API\_BASE\_URL, // set in .env file  headers: {  'Content-Type': 'application/json',  },  withCredentials: true,  });  export default apiClient;  const apiRoutes = {  users: '/users',  userById: (id: string) => `/users/${id}`,  posts: '/posts',  postById: (id: string) => `/posts/${id}`,  };  export default apiRoutes;  import apiClient from './apiClient';  type Method = 'GET' | 'POST' | 'PUT' | 'DELETE';  interface ApiRequest {  path: string;  method: Method;  data?: any;  params?: any;  }  const apiHandler = async ({ path, method, data, params }: ApiRequest) => {  try {  const response = await apiClient.request({  url: path,  method,  data,  params,  });  return response.data;  } catch (error: any) {  console.error('API Error:', error);  throw error.response?.data || error.message;  }  };  export default apiHandler;  import apiHandler from '@/services/apiHandler';  const useApi = () => {  const callApi = async (path: string, method: 'GET' | 'POST' | 'PUT' | 'DELETE', data?: any, params?: any) => {  return await apiHandler({ path, method, data, params });  };  return { callApi };  };  export default useApi;  'use client';  import { useEffect } from 'react';  import useApi from '@/hooks/useApi';  import apiRoutes from '@/services/apiRoutes';  const UserList = () => {  const { callApi } = useApi();  useEffect(() => {  const fetchUsers = async () => {  try {  const users = await callApi(apiRoutes.users, 'GET');  console.log(users);  } catch (err) {  console.error(err);  }  };  fetchUsers();  }, []);  return <div>User List</div>;  };  export default UserList;  // pages/\_app.tsx  import { Toaster } from 'react-hot-toast';  import type { AppProps } from 'next/app';  function MyApp({ Component, pageProps }: AppProps) {  return (  <>  <Toaster position="top-right" reverseOrder={false} />  <Component {...pageProps} />  </>  );  }  export default MyApp;  // pages/\_app.tsx  import { Toaster } from 'react-hot-toast';  import type { AppProps } from 'next/app';  function MyApp({ Component, pageProps }: AppProps) {  return (  <>  <Toaster position="top-right" reverseOrder={false} />  <Component {...pageProps} />  </>  );  }  export default MyApp;  import { useState } from 'react';  import apiHandler from '@/services/apiHandler';  import toast from 'react-hot-toast';  const useApi = () => {  const [loading, setLoading] = useState(false);  const [error, setError] = useState<any>(null);  const callApi = async (  path: string,  method: 'GET' | 'POST' | 'PUT' | 'DELETE',  data?: any,  params?: any,  showSuccessToast?: string,  showErrorToast?: boolean  ) => {  setLoading(true);  setError(null);  try {  const result = await apiHandler({ path, method, data, params });  if (showSuccessToast) {  toast.success(showSuccessToast);  }  return result;  } catch (err: any) {  setError(err);  if (showErrorToast !== false) {  toast.error(err.message || 'Something went wrong');  }  throw err;  } finally {  setLoading(false);  }  };  return { callApi, loading, error };  };  export default useApi; |