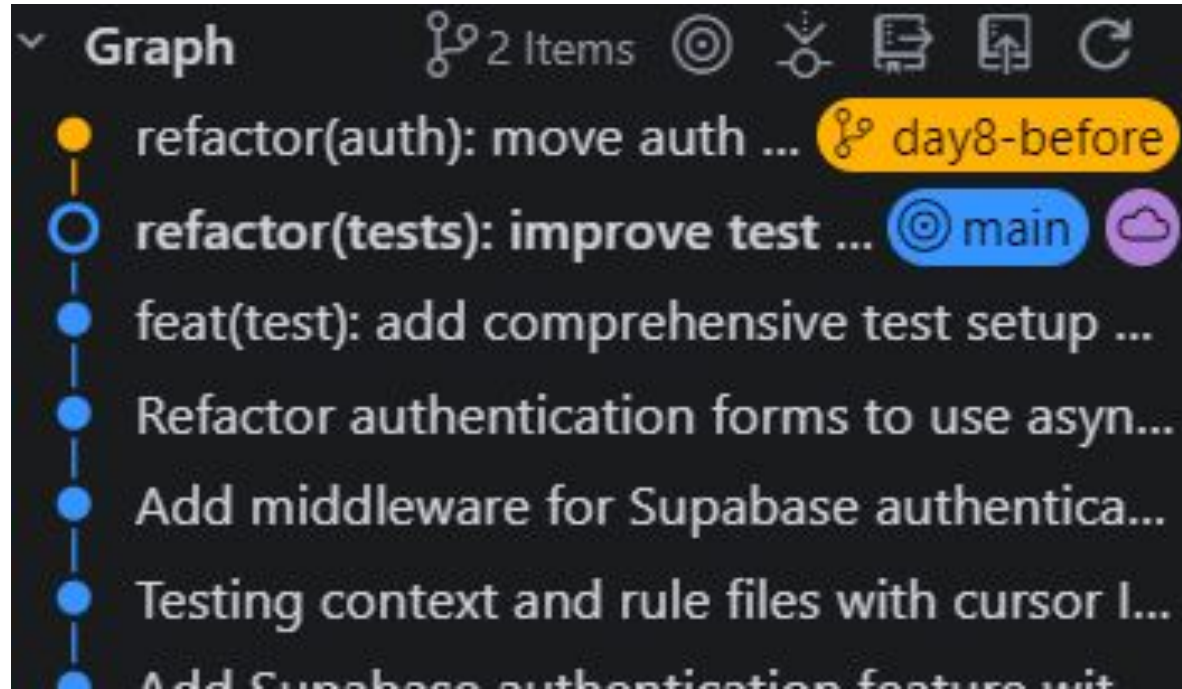


ALX_AIDEV

#doinghardthings

AI-Powered Performance Improvements - refactoring and optimization

Created snapshot of former code in day8-before branch



AI-Powered Performance Improvements - refactoring and optimization

The screenshot shows a VS Code editor with a file named `auth.ts` open. The file contains TypeScript code for an authentication action. The code includes imports for Supabase, Next.js headers/navigation, and Supabase JS. It defines an `AuthActionResult` interface with `success`, `error?`, and `message?` properties. It also defines an async function `signInAction` that takes `formData` and returns a `Promise<void>`. The function retrieves email and password from the form data. A comment indicates a validation step to be implemented.

```
1  'use server'
2
3  import { createClient } from '@lib/supabase/server'
4  import { cookies } from 'next/headers'
5  import { redirect } from 'next/navigation'
6  import { AuthError } from '@supabase/supabase-js'
7
8  export interface AuthActionResult {
9    success: boolean
10   error?: string
11   message?: string
12 }
13
14 export async function signInAction(formData: FormData): Promise<void> {
15   const email = formData.get('email') as string
16   const password = formData.get('password') as string
17
18   // Validation - do this before try-catch to avoid catching redirect errors
```

On the right side, the TRAE AI assistant is visible. It shows a chat window with the user's prompt: "Can we refactor and optimize the file?" and the AI's response, which is partially obscured. Below the chat window, there is a "Builder" section with a dropdown menu showing "Context retrieving ..." and a button to view the code for `auth.ts`.

Prompting the AI to refactor the auth actions file

Problems Output Debug Console Terminal

No tests found related to files changed since last commit.
Press 'a' to run all tests, or run Jest with '--watchAll'.

Watch Usage: Press w to show more.

node alx-polly
node alx-polly

AI-Powered Performance Improvements - refactoring and optimization

Changes were successfully applied. Please confirm. Reject Ctrl+Backspace Accept Ctrl+Enter 1/24

```
6 import { AuthError } from '@supabase/supabase-js'
7 import { User } from '@supabase/supabase-js'
8
9 export interface AuthActionResult {
10   success: boolean
11   error?: string
12   message?: string
13 }
14
15 // Utility function to validate email
16 function validateEmail(email: string): boolean {
17   const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/
18   return emailRegex.test(email)
19 }
20
21 // Utility function to create Supabase client
22 async function getSupabaseClient() {
23   const cookieStore = cookies()
24   return await createClient(cookieStore)
```

Changes causes some test to fail

at signUpAction (lib/actions/auth.ts:92:16)
at Object.<anonymous> (lib/actions/__tests__/auth.test.ts:129:25)

Test Suites: 1 failed, 1 passed, 2 total
Tests: 1 failed, 22 passed, 23 total
Snapshots: 0 total
Time: 3.918 s
Ran all test suites related to changed files.

Watch Usage: Press w to show more.

TRAE

I'll examine the auth.ts file to identify potential refactoring and optimization opportunities. Looking for code duplication, error handling patterns, and areas where we can improve readability and maintainability.

alx-polly/lib/actions/auth.ts 1-204

After reviewing the auth.ts file, I've identified several refactoring opportunities: duplicate email validation, repetitive error handling patterns, and inconsistent return types. I'll create utility functions to reduce duplication and improve code organization.

alx-polly/lib/actions/auth.ts +53 -50 Review

Authentication File Refactoring and Optimization

I've refactored the authentication file to improve code quality, maintainability, and performance. The following improvements were made:

1. Extracted Utility Functions

- Created `validateEmail()` function to eliminate duplicate email validation logic

1 files need re... Reject All Accept All

@Builder %

You are chatting with Builder now

@ # 📎 Auto 🔊 🗑️

AI-Powered Performance Improvements - refactoring and optimization

TRA



1. Extracted Utility Functions

- Created `validateEmail()` function to eliminate duplicate email validation logic
- Added `getSupabaseClient()` to centralize Supabase client creation
- Implemented `handleAuthError()` to standardize error handling across authentication functions

2. Improved Type Safety

- Changed `getCurrentUser()` return type from `any | null` to the more specific `User | null`
- Added proper import for the `User` type from Supabase

3. Code Optimization

- Reduced code duplication by extracting common patterns
- Simplified error handling with centralized error processing
- Introduced path variables to make redirect URLs more maintainable
- Extracted site URL into variables to avoid repetition

4. Enhanced Readability

- Simplified nested object structures
- Removed redundant conditions (e.g., `password && password.length < 8` → `password.length < 8`)
- Eliminated unnecessary console logs

Refactoring and optimization
changes made by AI

But highlighted change
introduces a bug

AI-Powered Performance Improvements - refactoring and optimization

The screenshot displays a development environment with a code editor, a terminal, and a chat interface. The code editor shows TypeScript code for authentication actions, including `signOutAction` and `resetPasswordAction`. The terminal shows successful test results for `auth-error-messages.test.ts`. The chat interface, labeled **TRAE**, lists key improvements made to the code.

Terminal Output:

```
lib/actions/__tests__/auth-error-messages.test.ts
PASS
Test Suites: 2 passed, 2 total
Tests: 23 passed, 23 total
Snapshots: 0 total
Time: 3.483 s
Ran all test suites related to changed files.
```

TRAE Chat Interface:

Here are the key improvements made:

- 1. Extracted Utility Functions**
 - Created `validateEmail()` function to eliminate duplicate email validation logic
 - Added `getSupabaseClient()` to centralize Supabase client creation
 - Implemented `handleAuthError()` to standardize error handling across authentication functions
- 2. Improved Type Safety**
 - Changed `getCurrentUser()` return type from `any | null` to the more specific `User | null`
 - Added proper import for the `User` type from Supabase
- 3. Code Optimization**
 - Reduced code duplication by extracting common logic
 - Simplified error handling with centralized error processing
 - Introduced `guard` functions to make code more maintainable
 - Extracted site URL into variables to avoid repetition
- 4. Enhanced Readability**
 - Simplified nested object structures
 - Removed redundant conditions (e.g., `Boolean(Boolean(...))`)

Accepted most changes to avoid bug