

CODE REVIEW

on a test project using Cypress e2e with allure reports

Content

Introduction.....	1
Package.json.....	2
README.md file.....	2
Configs: tsconfig.json, cypress.config.ts, config.config.ts	2
Pages: loginPage.ts	3
Test: login.test.ts + fixtures + support	3
CyAllureRun.yml	4
General conclusions.....	4

Introduction

Review on Cypress + Allure reports project, simple login procedure and testing logging in using valid and invalid credentials. This code review is done file by file with comments and observations from my side.

Package.json

- Observations:

String in the devDependencies:

`"@types/node": "^18.0.5",`

- - The current version of node is 22.11.0. Some functionality may not work properly.

`"scripts": "clear": "rm -r reports/** || true"`

- - The clear script is not supported on windows. Perhaps, it should be replaced with another one, universal for all systems, or write another script for Windows.

`"typescript": "^4.7.4"`

- Current version is 5.7.2. But I don't think it has any effect on the functionality for this project.

README.md file

- There is no information about system requirements and configuration. Perhaps, it would be good to add versions of Node.js, TypeScript, Cypress.
- The instruction "Change 'Username' and 'Password'" is not very clear. (This data is taken from the fixtures.json file.)
- There are no instructions on how to run allure reports. (maybe, this is already clear by default).

Configs: tsconfig.json, cypress.config.ts, config.config.ts

- - Config.config & cypress.config - matches the settings that can be defined in cypress.config.ts. Using two configs can cause confusion.

- Relative references with two dots (../) can cause problems on different OSes.
- It may be preferable to add an explanation as to why these particular timeouts were chosen.

Pages: loginPage.ts

LoginPage.ts:

- Methods (*“public launchApplication(), public login(emailId: string, password: string)”*) in the “Login Page” class use the modifier `public` for better readability and understanding the scope of the method.
- Variable types are explicitly specified, which is an understandable practice when using TypeScript.
- *public launchApplication(){ cy.visit('/') }*. When this method is called, the program will reference the base url specified in `config.config.ts`.

Test: login.test.ts + fixtures + support

- The code is written following best practices.
- The following error occurred when attempting to launch:

There were multiple support files found matching your `supportFile` pattern.

Your supportFile is set to: `cypress/support/e2e.{js,jsx,ts,tsx}`

We found the following files:

- `F:/Work_Code_Review/cypressAllure/cypress/support/e2e.js`
- `F:/Work_Code_Review/cypressAllure/cypress/support/e2e.ts`

Please remove or combine the support files into a single file.

Which required renaming or removing one of the files. The project used TypeScript, the solution was to rename the file `e2e.js` to `e2e-helper.js` (also could be just removed). After that, Cypress would load.

- When trying to run the test files, the test loops in an infinite reload. This error may be due to the fact that the `baseUrl` site is currently down. It could also possibly be due to the server redirecting requests incorrectly. Changing the `baseUrl` to the root address did not give any improvement.

CyAllureRun.yml

There is a difference in the versions used in package.json and CyAllureRun.yml file (matrix: node-version: [14.17.0]). That is, the code is written on version 18, but it is executed on version 14. This could be a potential problem, but it is not relevant for this project.

General conclusions

We have presented a project testing a rather simple function - login. The code seems readable, variables and methods are named descriptively.

The essence of the tests is to test login using valid/non-valid user data for login from fixtures and when entering "manually".

It was not possible to view the details of errors of failure of 5 out of 6 tests. No solutions were found to fix the error of cyclic page reloading when running the test. No obvious alternatives to the code writing variant were found, and the found additions were not significant, or there was a question about the expediency of adding them at all.

As a result of doing the review:

- Learned the basic vocabulary of TypeScript.
- What script granularity, new scripts.
- About the concept of fixtures.

There were some difficulties in performing this review due to the lack of sufficient knowledge about the architecture, possible implementations, some comparative base, thanks to which I could make a relative assessment of the quality of the project, what could be removed, and what could be added or done differently. A lot of video and text material was watched, and many requests were made to ChatGPT in order to understand the very concept of Code Review, as well as the technical details needed to understand the content of the project.