Asana Login Verification Test Write-Up

Table of Contents

Task Overview

Conduct data-driven tests via playwright for multiple test scenarios for the Asana application.

Implementation Details

File Locations

There are six tests in total and the data for each test is stored in a JSON object *testCases* inside the *./utils/testDataBank.js* file.

The data that *testCases* itself is composed of is imported from *./utils/fieldValidationTextValues.js* and each variable is stored as a *const* variable so that proper traceability is ensured and that the test suite is devoid of “magic-numbers” that can cause future confusion if the page text values are updated and there is no traceability to what goes where.

All **current** tests are contained inside *./tests/loginVerificationTests.spec.js* and are solely from that file.

Test Structure

All six tests are data-driven and run in parallel with one another.
Each test follows the same structure and flow.

Test Data Example

This is example data from test case #1.
Each test has these same entries with varying values.

```
TEST_CASE_ID_IDENTIFIER: 1,
TEST_CASE_NAME_IDENTIFIER: "Test Case 1",
TEST_CASE_LEFTNAV_IDENTIFIER: textBank.CROSS_FUNCTIONAL_PROJECT,
TEST_CASE_COLUMN_IDENTIFIER: textBank.TO_DO_COLUMN,
TEST_CASE_TASK_CARD_TITLE_IDENTIFIER: textBank.DRAFT_PROJECT_TASK,
TEST_CASE_TAGS_IDENTIFIER: [
        textBank.NON_PRIORITY_TAG,
        textBank.ON_TRACK_TAG
],
TEST_CASE_BOARD_CARD_TASK_ID: "1205367008167080"
```

beforeEach

Each test starts with an automated login to the Asana platform with data that is imported from *./utils/loginCredentials.js*. If multiple users are required in the future for testing these credentials can be expanded upon and be switched depending on user input.

test.step - Navigation

Each test navigates to the locator on the page that has a label of the current test cases' leftnav identifier.
Once this locator it is found and the test advances to the next step.

test.step – Card & Column Verification

There are four locators that are grabbed on the page and are found by their respective identifiers:

- columnLocator
- cardTextLocator
- columnLocatorAncestor
- cardTextLocatorAncestor

All four locators are then validated to be existent and visible on the page.
Though not directly stated as necessary to check by the client it was deemed necessary to utilize the ancestor locators to ensure that both column and card could be **automatically verified** to be inside the same column within the page, and not just present anywhere on the page.
Once all four locators are verified and visible the test advances to the final step.

test.step – Tag Verification

At the beginning of this step the card locator from the previous step is once again utilized, however in this step it is extracted by using the TASK_ID value as it is much more resilient to future change then retrieving the card by text title alone as a common function of these cards are to change their title names.
The TASK_ID will only be overwritten if the card itself as a whole is deleted.

This allows for nested locators and a much more direct extraction of the appropriate tag locators for the card in question.
These tag locators are then iterated through and verified to display their appropriate text when hovered upon.
Error handling in this case has been added and will throw a descriptive error in the event that a tag has text that is not approved of in the tag list.

The TASK_ID was not utilized in the previous step as Asana required that the text value be used for any interaction with the column.
During future meetings we will inform them of the possibility of using locators based on ids.

Once all tags are verified the test has concluded and we begin to clean up after the test.

afterEach

Though not explicitly asked for in their requirements, it was determined that it would be best to reduce load on repeated logins by properly logging out after each test.

We navigate to the user settings and successfully find the log out option and conclude the test after logging out of Asana.

Challenges and Solutions

Excessive Nesting

It was noted that throughout the application there were extreme levels of nesting of *div* elements that very rarely added functionality.
It is advised in the next meeting that we suggest reduced div usage.
An example would be taking the *div* elements with *class: draggable* and adding them to their children's *classList*.

Lack of IDs

It was noted that throughout the application very rarely were ids used.
It is advised in the next meeting that we suggest implementation of unique id identifiers for page elements.

Repeated Login Lockout

It was noted that Asana's system would frequently lock out the testing team due to the repeated login nature that was required of our tests.
It is advised in the next meeting that we suggest that special permissions are given to test accounts we use to prevent this lockout.

Requirements Constraints

It was noted that the client requirements sometimes conflicted with potentially much more stable options of gathering page locators.
It is advised in the next meeting that we suggest a lax on locator gathering requirements and that we suggest alternative methods such as seen with the inclusion of the TASK_ID field.

Results

Test Run Outcomes

Test runs are varying with flakiness observed with the wait times in some of the tests.
All tests will eventually pass if allowed to retry until the application can load it quickly enough. Increasing the wait times does not seem to help.
It appears that there is more flakiness the more the tests are run in parallel, and of these Firefox is the most volatile, with Chromium being the least volatile.

Examples of these test runs can be found in the *./TestReports* directory.
AllTestsPassed.html shows an example of it succeeding the first time.
AllTestsEventuallyPassedSomeFlaky.html shows an example of flakiness.

These tests were run and recorded one after another with no changes to the codebase.

Failures

As mentioned above some tests can fail the first time around and succeed after a retry.
This may be due to excessive parallelism causing stress on the application with some lockouts or taking too long.
Attempting more direct locators did not seem to remedy this issue.

Recommendations

On Tested Features

- Reduce *div* usage and unnecessary nesting
- Increase usage of *id* | *data-test-id* | *test-id* | …

On Testing Process

- Allow for tester flexibility in gathering of locators by more reliable means
- Allow special permissions for test accounts to prevent lockout

Tester Signature: Joseph Hegg