Web Automation

Technologies

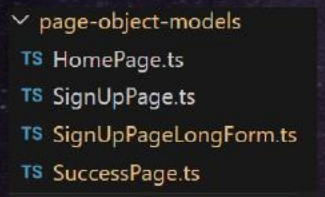
Playwright (latest), Node 20, Typescript

Pros:
- Playwright, an E2E language that I know how to use, very flexible, built in dynamic waiting for selectors/locators
- Node 20, it's simply required to orchestrate Playwright's project. It's good to be up to date in bug fixes and vulnerabilities
- Typescript, it makes Javascript typed and in a way, more similar to Java/Selenium. The types will add an extra layer of dev security for the developers when working on the project

Potential risks:
- Playwright, multi-platform support is not built in natively and has to be added by the developers for example integrating browserstack
- Node 20, it has to stay up to date to avoid vulnerabilities and bugs and maximize compatibility with playwright, an extra layer of potential failure in the project
- Typescript, added complexity to the project, some developers are not familiar with it

Best Practices

P.O.M.



web_automation > page-object-models > SuccessPage.ts > ...

```ts
import { expect, type Locator, type Page } from '@playwright/test';

export class SuccessPage {
  readonly page: Page;
  readonly pageUrl: string;
  readonly homePageUrl: string;
  readonly successMessage: Locator;
  readonly successMessageText: string;

  constructor(page: Page) {
    this.page = page;
    this.pageUrl = 'https://automationexercise.com/account_created';
    this.homePageUrl = 'https://automationexercise.com';
    this.successMessage = this.page.locator('h2[data-qa="account-created"]');
    this.successMessageText = 'Account Created!';
  }

  async goto() {
    await this.page.goto(this.pageUrl);
  }

  async verifySuccessMessage() {
    expect(await this.successMessage.textContent()).toBe(this.successMessageText);
  }

  async goBackToHomePage() {
    await this.page.goto(this.homePageUrl);
  }

}
```

Fixtures

web_automation > utils > BlockRequests.ts > ...

```ts
import type { Page } from '@playwright/test';

export class BlockRequests {
  constructor(public readonly page: Page) {
    this.execute();
  }

  async execute() {
    await this.page.route('**/pagead2.googlesyndication.com/**', async route => {
      await route.abort();
    });
  }
}
```

web_automation > tests > signupLoginExercise.spec.ts > ...

```ts
import { test, expect } from '@playwright/test';
import { HomePage } from '../page-object-models/HomePage';
import { SignUpPage } from '../page-object-models/SignUpPage';
import { SignUpPageLongForm } from '../page-object-models/SignUpPageLongForm';
import { SuccessPage } from '../page-object-models/SuccessPage';
import { BlockRequests } from '../utils/blockRequests';

test.beforeEach(async ({ page }) => {
  // for now just blocks the cookie consent popin
  const blockUndesiredRequests = new BlockRequests(page);
});

test('getting started should contain table of contents', async ({ page }) => {
  const homePage = new HomePage(page);
  const signUpPage = new SignUpPage(page);
```

Utils

web_automation > utils > helpers.ts > ...

```ts
export const generateTestEmail = (): string => `qa_testing_exercise${dateGenerator()}${rando
const dateGenerator = (): string => {
  const currentDate = new Date();
  return currentDate.toLocaleDateString().replace(/\//g,"-")
}

export const generateFinalEmail = (): string => `qa_testing_exercise@yopmail.com`

export const randomString = (length: number): string => {
  const characters = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789';
  let result = '';
  for (let i = 0; i < length; i++) {
    result += characters.charAt(Math.floor(Math.random() * characters.length));
  }
  return result;
}
```

```ts
import { generateTestEmail, generateFinalEmail, randomString } from '../utils/helpers';
  constructor(page: Page) {
    this.page = page;
    this.pageUrl = 'https://automationexercise.com/login';
    this.signUpNameText = `Jacint Iglesias Casanova ${randomString(5)}`;
    this.signUpEmailText = generateTestEmail();
```

User Api

Technologies

Postman

Pros:
Very user friendly program that I know how to use

Potential risks:
Final user needs the program to analize the exported collection

This API exercise has been created and executed using Postman. The directory of said project can be found in the directory called `user_api`.

Notes about the API project

You can find the `SQLI-test.postman_collection.json` file within the project's directory. This file is merely an export of the Postman collection.

Worth noting, perhaps is that this API took the POST (for user creation) and GET (for user retrieval by email key) parameters through `form-data` input instead of the most usual way which would be through body's JSon.

The main parameter for the API to set users as unique and thus avoid user duplication is the `email` field. I've taken this into consideration and I've attached a new email that I haven't used just so it will run fine the first time it'll be called.

user_api > {} SQLI-test.postman_collection.json > ...


```
{
  "info": {
    " postman_id": "ba0bc5d5-7a98-4110-9846-401c36bed647",
    "name": "SQLI-test",
    "schema": "https://schema.getpostman.com/json/collection/v2.1.0/collection.json",
    "_exporter_id": "12311009"
  },
  "item": [
    {
      "name": "Create User",
      "request": {
        "method": "POST",
        "header": [],
        "body": {
          "mode": "formdata",
          "formdata": [
            {
              "key": "name",
              "value": "John",
              "description": "\n",
              "type": "text"
            },
            {
              "key": "email",
              "value": "johnhasareallylonganduniqueemail1986@yopmail.com",
              "type": "text"
            },
            {
              "key": "password",
              "value": "ThisIsJohn'sPassword68$",
              "type": "text"
            },
            {
              "key": "title ",
              "value": "Mr",
              "type": "text"
            },
            {
              "key": "birth_date",
              "value": "10/10/1986",
              "type": "text"
            },
            {
              "key": "birth_month",
              "value": "10",
              "type": "text"
            },
```