Automating API Testing Using Playwright

Assigment:1

Test Case 1: Generate OAuth Token

Objective: Verify that a valid OAuth token can be generated.

Preconditions:

- Valid client credentials (`client_id`, `client_secret`, `username`, `password`).

Steps:

1. Send a POST request to the token endpoint with the required headers and form data.
2. Capture the response and validate the status code.
3. Verify that the response contains `access_token`, `instance_url`, and `token_type`.

Expected Results:

- Status code is `200`.
- The response body contains valid values for `access_token`, `instance_url`, and `token_type`.

Test Case 2: Create a New Case

Objective: Validate that a new lead can be created in Salesforce using the generated token.

Preconditions:

- A valid OAuth token is available.
- Faker is used to generate test data dynamically.

Steps:

1. Generate the token using the previous test case.
2. Send a POST request to the Salesforce API endpoint with:
    - Authorization header: Bearer token.
    - Body containing Status and Origin
3. Capture the response and validate the status code.
4. Verify that the response contains a valid `id`.

Expected Results:

- Status code is `201` (created).
- Response contains the `id` of the newly created resource.

Test Case 3: Retrieve Case Details

Objective: Ensure that the created case's details can be fetched successfully.

Preconditions:

- A lead has been created, and its `id` is available.

Steps:

1. Use the `id` of the created case from the previous test case.
2. Send a GET request with the authorization header.
3. Capture the response and validate the status code.

Expected Results:

- Status code is `200`.

Test Case 4: Update Case Details

Test Steps to update the case
1. Login to https://login.salesforce.com
2. Click on toggle menu button from the left corner
3. Click View All and click Sales from App Launcher
4. Click on Cases tab visible or select from more.
5. Click on the Dropdown icon and select Edit from the case you created by referring "case number"
6. Update Status as Working
7. Update Priority to low
8. Update Case Origin as Phone
9. Update SLA violation to No
10. Click on Save and Verify Status as Working

Expected result:
Case is Edited Successfully and Status is working

TestCase 5: Test Case: Delete Case in Salesforce

Objective: Verify that a case can be deleted from Salesforce using the API.

Preconditions:

- A lead has been created, and its `id` is available.

Steps:

4. Use the `id` of the created Case from the previous test case.
5. Send a Delete request with the authorization header.
6. Capture the response and validate the status code.

Expected Results:

- Status code is `204`.

Assigment:2

Test Case 1: Generate OAuth Token

Objective: Verify that a valid OAuth token can be generated.

Preconditions:

- Valid client credentials (`client_id`, `client_secret`, `username`, `password`).

Steps:

4. Send a POST request to the token endpoint with the required headers and form data.
5. Capture the response and validate the status code.
6. Verify that the response contains `access_token`, `instance_url`, and `token_type`.

Expected Results:

- Status code is `200`.
- The response body contains valid values for `access_token`, `instance_url`, and `token_type`.

Test Case 2: Create a New Contact

Objective: Validate that a new Contact can be created in Salesforce using UI.

Steps to create a New Contact

1) Launch the app
2) Click Login
3) Login with the credentials
4) Click on Global Actions SVG icon
5) After clicking Global Actions SVG icon, Click 'New Contact'.
6) Pick Salutation
7) Enter First Name
8) Enter Last Name
9) Enter email
10) Create a New Account for Account Name
11) Enter account name as 'Credits' and save
12) Click and save
13) Verify contact using Unique name and print the name

Test Case 3: Retrieve all Contacts Details

Objective: Ensure that the created contacts details can be fetched successfully.

Preconditions:

- A lead has been created, and its `id` is available.

Steps:

1. Send a GET request with the authorization header.
2. Capture the response and validate the status code.
3. You will get the response with Name and id .Fetch it

Expected Results:

- Status code is `200`.

Steps to Update the created Contact via Postman

Test Case: Update Contact Details Using a PATCH Call

Objective: Validate that the `PATCH` method can be used to update specific fields of an existing contact in Salesforce without modifying other fields.

Preconditions:

- A valid OAuth token is available.
- A lead is created, and its `id` is accessible.

Steps:

1.Send a `PATCH` request to the `/sobjects/Contact/{id}` endpoint with the following:

- **Authorization Header**: `Bearer <access_token>`
- **Content-Type**: `application/json`
- **Payload :** Phone ,Email, Title ,Department

2. Validate Response:

- Check the response status code.
- Verify that the status code indicating the update was successful

---

TestCase 5: Test Case: Delete Lead in Salesforce

Objective: Verify that a lead can be deleted from Salesforce using the API.

Preconditions:

- A contact has been created, and its `id` is available.

Steps:

1. Use the `id` of the created contact from the previous test case.

2. Send a Delete request with the authorization header.

3. Capture the response and validate the status code.

Expected Results:

- Status code is `204`.

Assignment- 3

Try implementing utility function for API integration with Playwright

Reference Code:

```
import test from "@playwright/test";
import { createResource, generateToken, getResource } from "./apiUtility";
import { faker } from "@faker-js/faker";

let lastname=faker.person.lastName()
let compname=faker.company.buzzNoun()
let id:any
let searchname:any
test.beforeEach(`SalesForce lead Generation`,async({page})=>{
   await generateToken(page.request)
   id= await createResource(page.request,lastname,compname)
   searchname= await getResource(page.request)
})
test(`Salesforce using ui data verification`,async({page})=>{
   await page.goto("https://login.salesforce.com/")
   await page.fill("#username","vidyar@testleaf.com")
   await page.fill("#password","Sales@123")
   await page.click("#Login")
   await page.waitForLoadState('load')
   await page.locator(".slds-icon-waffle").click();
   await page.locator("//button[text()='View All']") .click()
   await page.getByPlaceholder("Search apps or items...").fill("Leads")
   await page.locator("//mark[text()='Leads']").click()
   await page.waitForLoadState('load')
   await page.getByPlaceholder("Search this list...").fill(searchname,{timeout:20000})
})
```