To set up **Typo.js** for spell-checking a webpage in a Playwright test and generate a report using **Allure**, follow these steps:

**Step 1: Install Required Packages**

Install Playwright and Allure Reporter:

From bash Terminal

npm install playwright @playwright/test allure-playwright

Set up Allure CLI for generating the report:

bash

npm install -g allure-commandline --save-dev

**Step 2: Install Typo.js**

Typo.js isn't available via npm, so download it directly from its [GitHub repository](https://github.com/cfinke/Typo.js). Save the typo.js file in your project directory.

**Step 3: Download Dictionary Files**

Download .aff and .dic files for your language (e.g., en\_US.aff and en\_US.dic) from the [Hunspell repository](https://github.com/wooorm/dictionaries) and place them in a dictionaries/ folder in your project.

**Step 4: Configure Playwright with Allure**

Create a playwright.config.js file to enable Allure reporting:

javascript

const { defineConfig } = require('@playwright/test');

module.exports = defineConfig({

reporter: [['allure-playwright']],

});

**Step 5: Create the Playwright Test**

Write a test script to spell-check webpage content using Typo.js and report the results to Allure.

**Test Script: spellCheck.test.js**

javascript

const { test, expect } = require('@playwright/test');

const fs = require('fs');

const path = require('path');

test.describe('Spell Check Test', () => {

const typoJsPath = path.resolve('./typo.js');

const affPath = path.resolve('./dictionaries/en\_US.aff');

const dicPath = path.resolve('./dictionaries/en\_US.dic');

let affContent, dicContent;

test.beforeAll(() => {

// Load dictionary files

affContent = fs.readFileSync(affPath, 'utf8');

dicContent = fs.readFileSync(dicPath, 'utf8');

});

test('Check spelling on webpage', async ({ page }) => {

await test.step('Navigate to webpage', async () => {

await page.goto('https://example.com'); // Replace with the URL to test

});

await test.step('Inject Typo.js', async () => {

const typoJsCode = fs.readFileSync(typoJsPath, 'utf8');

await page.addScriptTag({ content: typoJsCode });

});

await test.step('Initialize Typo.js', async () => {

await page.evaluate(({ affContent, dicContent }) => {

window.typo = new Typo('en\_US', affContent, dicContent);

}, { affContent, dicContent });

});

const pageText = await test.step('Extract page text', async () => {

return page.evaluate(() => document.body.innerText);

});

const misspelledWords = await test.step('Check for misspelled words', async () => {

return page.evaluate((text) => {

const words = text.split(/\s+/);

return words.filter((word) => !window.typo.check(word));

}, pageText);

});

// Attach results to Allure report

await test.step('Generate Allure report', async () => {

test.info().attach('Misspelled Words', {

body: JSON.stringify(misspelledWords, null, 2),

contentType: 'application/json',

});

});

// Assert no spelling errors

expect(misspelledWords).toEqual([]);

});

});

**Step 6: Run the Test**

Run the Playwright test suite:

From bash Terminal

**npx playwright test**

**Step 7: Generate Allure Report**

1. Generate the Allure results: From bash Terminal

**npx allure generate allure-results --clean**

1. Serve the report: From bash Terminal

**npx allure open**

This will open a web interface showing detailed test results, including the misspelled words identified during the test.

**Explanation of Steps**

1. **Typo.js Setup**: The Typo.js library and dictionary files are injected into the webpage for spell-checking.
2. **Spell Checking**: The webpage text is extracted, split into words, and checked against the dictionary.
3. **Allure Integration**: Misspelled words are attached to the Allure report for better visibility.
4. **Assertions**: The test fails if any misspelled words are found, ensuring quality control.

**Customizations**

* **Multiple Pages**: Extend the test to iterate over multiple URLs.
* **Language Support**: Use different .aff and .dic files for various languages.
* **Dynamic Suggestions**: Include Typo.js's suggest() method to add correction suggestions to the report.