

Playwright Setup Guide: Basic framework and Write your first test case

What is Playwright?

Playwright is a cross-browser automation framework by Microsoft for end-to-end testing of web applications. It supports major browsers- Chromium (Chrome, Edge), Firefox, and WebKit (Safari)- with a unified API.

Key Features:

- **Cross-Browser Support:** Test across multiple browsers with a single API.
- **Language Flexibility:** Write tests in JavaScript, TypeScript, Python, C#, and Java.
- **Headless Mode:** Run the tests without a GUI for speed and efficiency.
- **Auto-Waiting:** Reduces the flaky tests by waiting for elements to be ready.
- **Screenshots & Videos:** Capture the test execution for better debugging.
- **Network Interception:** Simulate and modify the network conditions.

Playwright simplifies testing, ensures reliability, and integrates seamlessly into CI/CD pipelines, making it a powerful choice for modern web application testing.

Prerequisites:

Before setting up Playwright, ensure you have **Node.js** and **Visual Studio** installed.

What is Node.js?

Node.js is a runtime environment that allows you to execute **JavaScript code** **server-side**, outside of a web browser.

Install Node:

Download Node.js®

Download Node.js the way you want.

[Package Manager](#) [Prebuilt Installer](#) [Prebuilt Binaries](#) [Source Code](#)

Install Node.js

v20.16.0 (LTS) ▾

 on

🍏 macOS ▾

 using

🔧 nvm ▾

```
1 # installs nvm (Node Version Manager)
2 curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.40.0/install.sh | bash
3
4 # download and install Node.js (you may need to restart the terminal)
5 nvm install 20
6
7 # verifies the right Node.js version is in the environment
8 node -v # should print `v20.16.0`
9
10 # verifies the right npm version is in the environment
11 npm -v # should print `10.8.1`
```

Installation Instructions (on MacOS)

Run the below mentioned commands or install the commands based on your OS:

installs nvm (Node Version Manager)

```
curl -o-
https://raw.githubusercontent.com/nvm-sh/nvm/v0.40.0/install.sh |
bash
```

download and install Node.js (you may need to restart the terminal)

```
nvm install 20
```

verifies the right Node.js version is in the environment

```
node -v # should print `v20.x.x`
```

verifies the right npm version is in the environment

```
npm -v # should print `10.x.x`
```

```
[priyankamadanlal@Priyankas-MBP ~ % node -v  
v20.8.1
```

```
[priyankamadanlal@Priyankas-MBP ~ % npm -v  
10.8.1
```

Download and install the Visual Studio from the below website:

[Visual Studio Code](#)

Setup Playwright:

1. Create an Empty Folder: Create a new directory for your project.
2. Open the Folder in Visual Studio Code: Open Visual Studio Code and navigate to the created folder.

Install Playwright Dependencies:

Run the below command to install the dependencies:

```
npm init playwright
```

This command sets up the Playwright environment and creates the necessary files for your end-to-end testing framework.

```

❯ priyankamadanlal@Priyankas-MBP qa-playwright-automation % npm init playwright

> npx
> create-playwright

Getting started with writing end-to-end tests with Playwright:
Initializing project in '.'
✓ Do you want to use TypeScript or JavaScript? · JavaScript
✓ Where to put your end-to-end tests? · tests
✓ Add a GitHub Actions workflow? (y/N) · true
✓ Install Playwright browsers (can be done manually via 'npx playwright install')? (Y
  · true

Initializing NPM project (npm init -y)...
Wrote to /Users/priyankamadanlal/workspace/qa-playwright-automation/package.json:

{
  "name": "qa-playwright-automation",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": ""
}

Installing Playwright Test (npm install --save-dev @playwright/test)...

added 3 packages, and audited 4 packages in 4s

found 0 vulnerabilities
Installing Types (npm install --save-dev @types/node)...

added 3 packages, and audited 7 packages in 1s

found 0 vulnerabilities
Writing playwright.config.js.
Writing .github/workflows/playwright.yml.
Writing tests/example.spec.js.
Writing tests-examples/demo-todo-app.spec.js.
Writing package.json.
Downloading browsers (npx playwright install)...

```

package.json Configuration:

Here's the configuration for your `package.json` file:

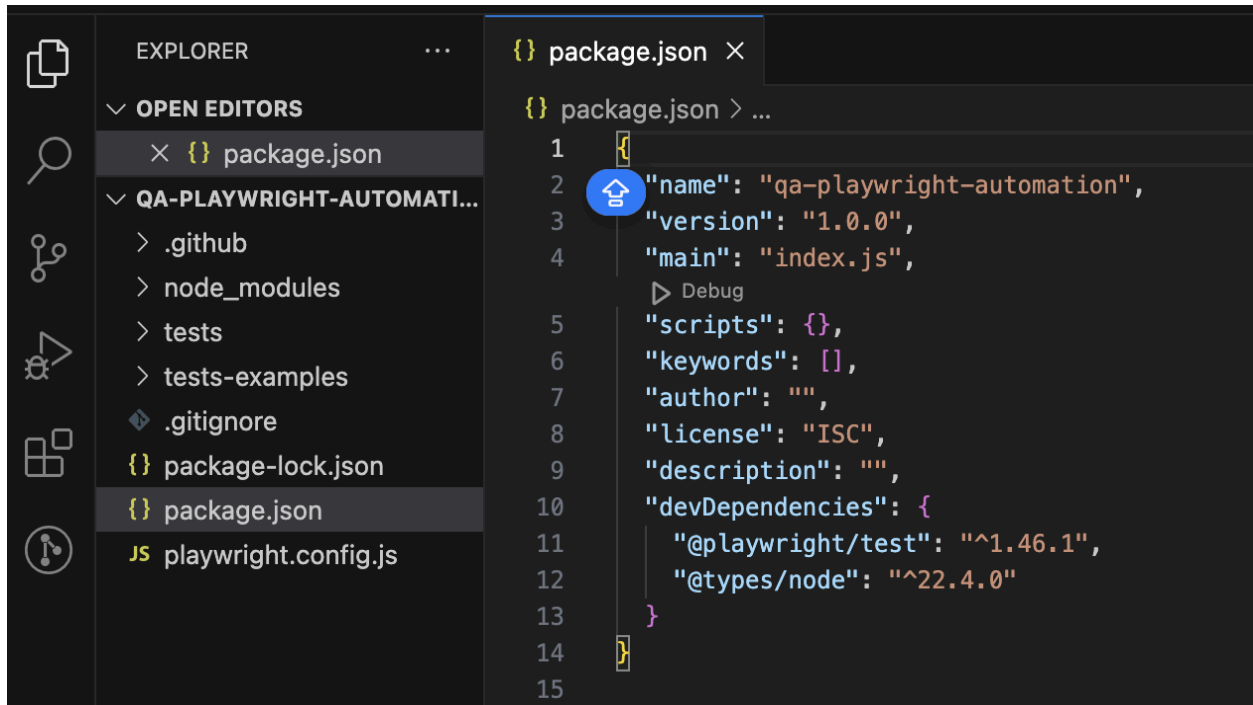
Code to copy:

```

{
  "name": "qa-playwright-automation",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {},
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": "",

```

```
"devDependencies": {
  "@playwright/test": "^1.46.1",
  "@types/node": "^22.4.0"
}
```



Note: This configuration includes Playwright and Node.js type definitions.

Playwright Configuration File:

A default [playwright.config.js](#) will be created with the following content:

```

JS playwright.config.js > [?] config > [?] reporter
1  // @ts-check
2  const { defineConfig } = require('@playwright/test');
3
4  const config = defineConfig({
5    testDir: './tests',
6    timeout: 30 * 1000, // Global timeout for tests
7    expect: {
8      timeout: 5000 // Timeout for individual assertions
9    },
10   reporter: [['html', { open: 'never' }]], // Configure HTML reporter
11   use: {
12     baseURL: 'https://opensource-demo.orangehrmlive.com/web/index.php/auth/login',
13     browserName: 'chromium',
14   },
15 });
16
17 module.exports = config;
18

```

Code to copy

```

// @ts-check
const { defineConfig } = require('@playwright/test');
const config = defineConfig({

  testDir: './tests',
  timeout: 30 * 1000, // Global timeout for tests
  expect: {
    timeout: 5000 // Timeout for individual assertions
  },
  reporter: [['html', { open: 'never' }]], // Configure HTML reporter
  use: {
    baseURL:
'https://opensource-demo.orangehrmlive.com/web/index.php/auth/login',
    browserName: 'chromium',
  },
});
module.exports = config;

```

Write your first test case:

Create a *.spec.js file under /tests folder:

Code to copy

```
const { test, expect } = require('@playwright/test');
test('Verify the page title of OrangeHRM login page', async ({ page }) => {
  await page.goto('/'); // This uses the baseURL
  console.log(await page.title());
  await expect(page).toHaveTitle("OrangeHRM");
});
```

Running Tests:

Running tests sequentially:

```
npx playwright test
```

Run tests in headed Mode:

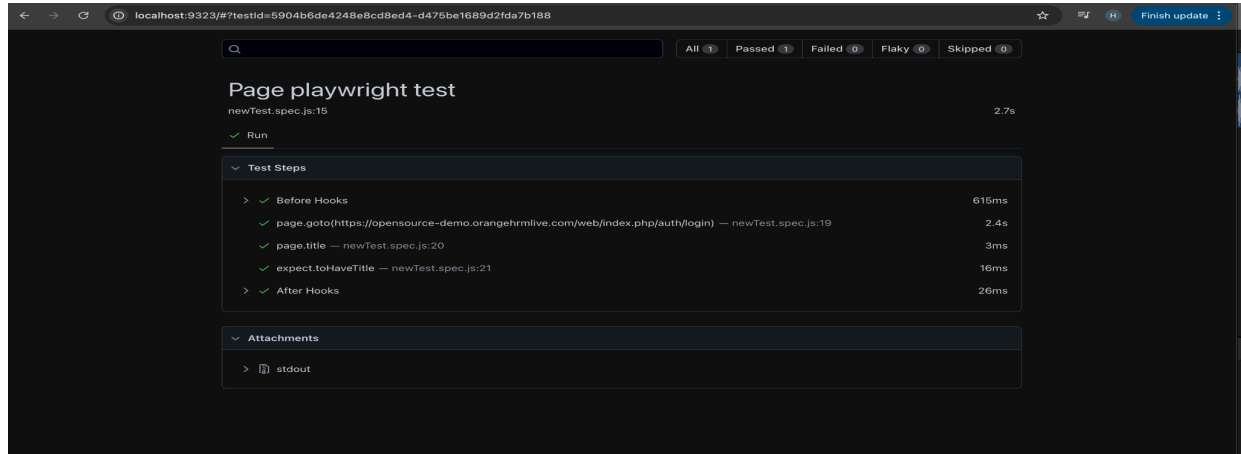
```
npx playwright test --headed
```

Viewing Reports:

Generate and View HTML Report:

```
npx playwright show-report
```

The above command opens the last HTML report showing passed and failed tests along with attachments.



Add Scripts to package.json

To simplify running Playwright commands, add these scripts to your package.json:

```
"scripts": {  
  "playwright": "playwright test",  
  "playwright:headed": "playwright test --headed",  
  "playwright:report": "playwright show-report"  
},
```

Run Playwright:

```
npm run playwright
```

Run in Headed Mode:

```
npm run playwright:headed
```

View Playwright Report:


```
npm run playwright:report
```

Test Passed Successfully!

```
Node.js v20.8.1
● priyankamadanlal@Priyankas-MBP qa-playwright-automation % npm run playwright:open

> qa-playwright-automation@1.0.0 playwright:open
> playwright test --headed

Running 1 test using 1 worker
newTest.spec.js:15:6 > Page playwright test
OrangeHRM
  1 passed (2.9s)

To open last HTML report run:

  npx playwright show-report
```

Git Repo: <https://github.com/prgupta1995/playwright-web-test>

Next, we'll discuss running the playwright tests in multiple browsers.

Keep learning and keep sharing!

Priyanka

Senior QA Engineer