Playwright Automation Guide - Complete

Complete Playwright Automation Guide

Introduction

Playwright is a modern test automation framework developed by Microsoft. It supports testing across Chromium, Firefox, and WebKit with a single API. It is widely adopted for its speed, reliability, and support for advanced automation scenarios.

What You Will Learn

- Installation and setup
- JavaScript basics (if new to programming)
- CLI test execution and debugging
- Locators and test organization
- Page Object Model (POM)
- UI and API automation
- Advanced Playwright topics

1. Installation and Setup

npm init -y

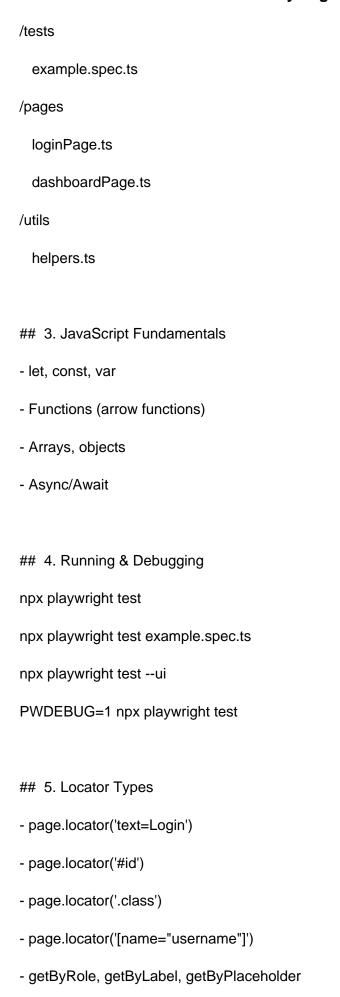
npm i -D @playwright/test

npx playwright install

npx playwright test --init

2. Project Structure

playwright.config.ts



- Advanced: :has-text(), :nth-match(), :scope

6. Assertions

- expect(locator).toBeVisible()

- expect(locator).toHaveText()

- expect(page).toHaveURL()
- expect(locator).toHaveAttribute()
7. Hooks
test.beforeEach(), test.afterEach(), etc.
8. Page Object Model (POM)
- Class-based abstraction
- Reusable locators and methods
9. API Testing
- request.post(), request.get()
- API mocking with route.fulfill()
10. Advanced Topics
- Fixtures, retries, timeouts
- Tags, parallel execution
- Docker container setup
11. Reporters
- HTML report, list, line

- Allure (optional setup)
12. Mobile Emulation
- use: {devices['Pixel 5'] }
13. Visual Testing
- toHaveScreenshot()
14. Practice Scenarios
- Login/logout
- Dropdowns, checkboxes, tables
- Uploads, tooltips, popups
- Sliders, date pickers, iFrames
15. CI/CD
GitHub Actions Workflow example for automation
16. Bonus (Optional Advanced Concepts)
- Multi-browser Execution
- Test Data Management
- Custom Helper Functions
- Allure Reporting Integration
- GitHub Secrets + CI Variables
- Test Video Recording on Failure
- Environment Switching (.env)
- Error Handling Strategies

- Accessibility Testing (a11y)
- Component Testing (React/Angular)

Summary

This guide prepares you to confidently write, run, and manage Playwright test frameworks for real-world applications. You are now job-ready and project-ready.

Happy Testing!