

To discuss this course and customizations:  
Call: +1 434-509-6890 or Email: [sales@cloudcontraptions.com](mailto:sales@cloudcontraptions.com)

# End-to-End Testing with Playwright

---

## Class Duration

14 hours of live training delivered over 2 days.

## Student Prerequisites

- Working knowledge of JavaScript or TypeScript
- Familiarity with how web applications work (HTML, DOM, HTTP)
- Basic experience with any testing framework is helpful
- No prior Playwright or end-to-end testing experience required

## Target Audience

This course is designed for developers and QA engineers responsible for end-to-end test suites - whether starting fresh with Playwright, migrating from Selenium or Cypress, or rescuing an existing suite that has grown slow and flaky. It suits full-stack developers who own quality for their features, dedicated test engineers building shared frameworks, and teams that want to take advantage of Playwright's newer agentic and AI-assisted capabilities without sacrificing suite reliability.

## Description

End-to-end tests have a reputation: slow, flaky, and abandoned six months after they were written. Playwright was designed to break that pattern, and this course teaches the philosophy that makes it work - auto-waiting, user-visible locators, and web-first assertions that retry until the page settles instead of failing on arbitrary timing. Participants build a real test suite from the ground up: structuring tests with fixtures for true isolation, choosing between page objects and role-based patterns, intercepting and mocking network traffic, and reusing authentication state so suites do not waste minutes logging in.

The second day focuses on running tests at scale and keeping them trustworthy: parallelism and sharding, debugging failures with the trace viewer, visual comparisons, cross-browser projects and mobile emulation, and CI integration with GitHub Actions. A dedicated module addresses flake reduction as an engineering discipline rather than a retry-button habit. The

To discuss this course and customizations:  
Call: +1 434-509-6890 or Email: [sales@cloudcontraptions.com](mailto:sales@cloudcontraptions.com)

course also covers Playwright's API testing capabilities and closes with its agentic side, current as of Playwright 1.60: the Playwright Agents workflow (planner, generator, healer) for AI-assisted test creation and repair, agentic video receipts, and how to review and maintain AI-generated tests so they meet the same standard as hand-written ones.

## Learning Outcomes

- Explain Playwright's auto-waiting model and why it eliminates most explicit waits and sleeps
- Write resilient tests using role-based locators and web-first assertions
- Structure suites with fixtures that guarantee test isolation and share setup without shared state
- Choose between page object models and role-based locator patterns, and apply each cleanly
- Intercept, mock, and assert on network traffic to isolate the frontend under test
- Reuse authentication state across tests and projects to keep suites fast
- Configure parallel workers, projects, and sharding to scale suites across machines
- Debug failures efficiently with the trace viewer, UI mode, and HTML reports
- Implement visual comparison tests and manage snapshots responsibly
- Run suites in CI with GitHub Actions, including artifacts, retries, and shard orchestration
- Diagnose and systematically eliminate the real causes of flaky tests
- Test across Chromium, Firefox, and WebKit, including mobile emulation
- Use Playwright's request context for API testing and hybrid UI-plus-API scenarios
- Apply Playwright Agents (planner, generator, healer) to generate and repair tests with AI assistance, and review the output critically

## Training Materials

Comprehensive courseware is distributed online at the start of class. All students receive a downloadable MP4 recording of the training.

## Software Requirements

- Node.js LTS (20 or later) installed locally
- Playwright 1.60 or later (`npm init playwright@latest`)

To discuss this course and customizations:  
Call: +1 434-509-6890 or Email: [sales@cloudcontraptions.com](mailto:sales@cloudcontraptions.com)

- Playwright browsers installed (Chromium, Firefox, WebKit)
- VS Code with the official Playwright Test extension recommended
- A GitHub account for the CI labs
- Git for version control

## **Training Topics**

### **Playwright Philosophy and Architecture**

- Why end-to-end suites fail and how Playwright's design responds
- Auto-waiting and actionability checks explained
- Browsers, contexts, and pages
- Selectors history: why user-facing locators won

### **Locators and Web-First Assertions**

- Role, label, text, and test-id locators
- Filtering and chaining locators
- Web-first assertions that retry until the page settles
- Soft assertions and custom expect messages
- Locator strategies that survive refactoring

### **Fixtures and Test Isolation**

- Built-in fixtures: page, context, browser, request
- Writing custom fixtures for setup and teardown
- Worker-scoped versus test-scoped fixtures
- Why isolation matters more than speed tricks

### **Structuring a Suite**

- Page object model: benefits and failure modes
- Role-based patterns as a lighter alternative
- Sharing helpers without hidden coupling
- Tagging, grepping, and organizing by feature

### **Network Interception and Mocking**

- Routing, fulfilling, and modifying requests
- Mocking APIs to isolate frontend behavior
- HAR recording and replay
- Asserting on outbound requests

To discuss this course and customizations:  
Call: +1 434-509-6890 or Email: [sales@cloudcontraptions.com](mailto:sales@cloudcontraptions.com)

### **Authentication State Reuse**

- Storage state: capture once, reuse everywhere
- Project dependencies and global setup
- Multiple roles and per-worker accounts
- Keeping secrets out of test code

### **Parallelism and Sharding**

- Workers, fully parallel mode, and serial escape hatches
- Projects for browsers and environments
- Sharding suites across CI machines
- Merging reports from sharded runs

### **Debugging with the Trace Viewer**

- Recording traces on failure and retry
- Reading the timeline: actions, snapshots, network, console
- UI mode for fast local iteration
- HTML reports and failure triage

### **Visual Comparisons**

- Screenshot assertions and snapshot management
- Masking dynamic regions and setting thresholds
- Cross-platform snapshot pitfalls and CI-generated baselines

### **CI Integration with GitHub Actions**

- The Playwright container and browser caching
- Uploading traces, videos, and reports as artifacts
- Retries, quarantine, and failure budgets
- Sharded workflows with report merging

### **Flake Reduction and Cross-Browser Testing**

- The real causes of flakiness: timing, state, environment
- Hunting flakes with --repeat-each and trace analysis
- Testing Chromium, Firefox, and WebKit from one suite
- Mobile emulation: viewports, touch, and device profiles

### **API Testing and AI-Assisted Generation**

- The request fixture for pure API tests
- Hybrid scenarios: seed via API, verify via UI
- Playwright Agents: planner, generator, and healer workflow



To discuss this course and customizations:  
Call: +1 434-509-6890 or Email: [sales@cloudcontraptions.com](mailto:sales@cloudcontraptions.com)

- Generating tests from a Markdown plan and healing failures
- Agentic video receipts and reviewing agent runs
- Holding AI-generated tests to human standards