Mastering Playwright

A Comprehensive Guide to Playwright Commands for Testing and Debugging

Testleaf
Always Ahead

Getting Started with Playwright

Playwright is a powerful tool for automating web applications, enabling you to perform end-to-end testing across various browsers. Below is a comprehensive guide to using Playwright, including installation, running tests, and configuring various options.

Installation

1. Install Playwright

Command:

```
npm install playwright
```

Explanation: This command installs Playwright in your project, allowing you to start writing tests. After running this command, Playwright will be available in your project.

2. Install Playwright with Browsers

Command:

```
npx playwright install
```

Explanation: This installs Playwright along with the required browsers—Chromium, Firefox, and WebKit. This is particularly useful for setting up a fresh Playwright project.

Running Tests

3. Run Tests

Command:

```
npx playwright test
```

Explanation: Runs all the Playwright tests in your project. You can execute this command from your project directory to run all your tests.

4. Run Tests for a Specific File

Command:

```
npx playwright test tests/example.spec.ts
```

Explanation: Executes tests from a specific test file. This is useful for focusing on a particular test file during development.

5. Run Tests in Headed Mode

Command:

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

Explanation: Runs the tests with the browser UI visible. This is helpful for debugging, as you can observe the browser actions directly.

6. Run Tests in Headless Mode

Command:

```
npx playwright test --headless
```

Explanation: Runs the tests without opening the browser UI, the default mode. This is ideal for CI/CD environments where UI is not needed.

Test Configuration

7. Run Tests with Timeout

Command:

```
npx playwright test --timeout=60000
```

Explanation: Sets a timeout of 60 seconds for each test to prevent them from hanging.

8. Run Tests in Parallel

Command:

```
npx playwright test --workers=4
```

Explanation: Runs tests in parallel using 4 workers, speeding up test execution on multi-core machines.

Reporting

9. Generate HTML Report

Command:

```
npx playwright test --reporter=html
```

Explanation: Generates an HTML report after running the tests, useful for reviewing detailed results.

10. Run Tests with a Custom Reporter

Command:

```
npx playwright test --reporter=json
```

Explanation: Outputs a JSON report, useful for further processing or integration with other tools.

Advanced Usage

11. Filter Tests by Name

Command:

```
npx playwright test -g 'login'
```

Explanation: Runs only the tests that match the provided name or tag, such as 'login'.

12. Run Tests in Specific Browser

Command:

```
npx playwright test --project=firefox
```

Explanation: Executes tests specifically using Firefox, helpful for testing browser-specific behaviors.

13. Capture Screenshot on Failure

Command:

```
npx playwright test --screenshot=only-on-failure
```

Explanation: Captures screenshots only when tests fail, providing visual feedback for debugging.

14. Run Tests with Debugging Info

Command:

```
npx playwright test --debug
```

Explanation: Enables detailed debugging output, useful for diagnosing issues during test execution.

15. Open a Browser with codegen

Command:

```
npx playwright codegen https://example.com
```

Explanation: Opens a browser and records interactions, generating test code based on your actions.

16. Install a Specific Browser

Command:

```
npx playwright install chromium
```

Explanation: Installs only the Chromium browser, useful if you need a specific browser for testing.

17. Create a Custom Browser Context

Command:

```
npx playwright test --context=browserContextName
```

Explanation: Runs tests using a specific browser context, useful for isolated browser sessions.

18. Generate a Trace for Debugging

Command:

```
npx playwright test --trace=on
```

Explanation: Generates a trace of test execution, aiding in debugging by analyzing steps performed during the test.

19. Run Tests with Retried Failures

Command:

```
npx playwright test --retries=3
```

Explanation: Retries each test up to 3 times if they fail, helping to reduce flaky test failures in CI environments.

20. Run Tests with Video Recording

Command:

```
npx playwright test --video=on
```

Explanation: Records a video of the test execution, useful for visually reviewing test failures.

These commands offer a wide range of functionalities to help you efficiently manage and execute tests using Playwright. Whether you're setting up a new project or refining your testing process, these tools can greatly enhance your testing capabilities.