Skip to main content

PlaywrightDocsAPI
Node.js

- Node.js
- Python
- Java
- .NET

Community
Search⌘K

- Getting Started
  - Installation
  - Writing tests
  - Generating tests
  - Running and debugging tests
  - Trace viewer
  - Setting up CI
- Getting started - VS Code
- Release notes
- Canary releases
- Playwright Test
  - Test configuration
  - Test use options
  - Annotations
  - Command line

- Emulation
    - Fixtures
    - Global setup and teardown
    - Parallelism
    - Parameterize tests
    - Projects
    - Reporters
    - Retries
    - Sharding
    - Timeouts
    - TypeScript
    - UI Mode
    - Web server
- Guides
    - Library
    - Accessibility testing
    - Actions
    - Assertions
    - API testing
    - Authentication
    - Auto-waiting
    - Best Practices
    - Browsers
    - Chrome extensions
    - Clock
    - Components (experimental)
    - Debugging Tests
    - Dialogs
    - Downloads
    - Evaluating JavaScript
    - Events
    - Extensibility
    - Frames
    - Handles
    - Isolation
    - Locators
    - Mock APIs
    - Mock browser APIs
    - Navigations
    - Network
    - Other locators
    - Page object models
    - Pages
    - Screenshots
    - Visual comparisons
    - Test generator
    - Trace viewer
    - Videos
    - WebView2
- Migration

- Integrations
- Supported languages

-
- Playwright Test
- Command line

On this page

Command line

Introduction

Here are the most common options available in the command line.

- Run all the tests

  ```
  npx playwright test
  ```

- Run a single test file

  ```
  npx playwright test tests/todo-page.spec.ts
  ```

- Run a set of test files

  ```
  npx playwright test tests/todo-page/ tests/landing-page/
  ```

- Run files that have `my-spec` or `my-spec-2` in the file name

  ```
  npx playwright test my-spec my-spec-2
  ```

- Run tests that are in line 42 in my-spec.ts

  ```
  npx playwright test my-spec.ts:42
  ```

- Run the test with the title

  ```
  npx playwright test -g "add a todo item"
  ```

- Run tests in headed browsers

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

- Run all the tests against a specific project

  ```
  npx playwright test --project=chromium
  ```

- Disable parallelization

  ```
  npx playwright test --workers=1
  ```

- Choose a reporter

```
npx playwright test --reporter=dot
```

- Run in debug mode with Playwright Inspector

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

- Run tests in interactive UI mode, with a built-in watch mode (Preview)

```
npx playwright test --ui
```

- Ask for help

```
npx playwright test --help
```

Reference

Complete set of Playwright Test options is available in the configuration file. Following options can be passed to a command line and take priority over the configuration file:

Option	Description
Non-option arguments	Each argument is treated as a regular expression matched against the full test file path. Only tests from the files matching the pattern will be executed. Special symbols like `$` or `*` should be escaped with `\`. In many shells/terminals you may need to quote the arguments.
`-c <file>` or `--config <file>`	Configuration file. If not passed, defaults to `playwright.config.ts` or `playwright.config.js` in the current directory.
`--debug`	Run tests with Playwright Inspector. Shortcut for `PWDEBUG=1` environment variable and `--timeout=0 --max-failures=1 --headed --workers=1` options.
`--fail-on-flaky-tests`	Fails test runs that contain flaky tests. By default flaky tests count as successes.
`--forbid-only`	Whether to disallow `test.only`. Useful on CI.
`--global-timeout <number>`	Total timeout for the whole test run in milliseconds. By default, there is no global timeout. Learn more about various timeouts.
`-g <grep>` or `--grep <grep>`	Only run tests matching this regular expression. For example, this will run `'should add to cart'` when passed `-g "add to cart"`. The regular expression will be tested against the string that consists of the project name, test file name, `test.describe` titles if any, test title and all test tags, separated by spaces, e.g. `chromium my-test.spec.ts my-suite my-test @smoke`. The filter does not apply to the tests from dependency projects, i.e. Playwright will still run all tests from project dependencies.
`--grep-invert <grep>`	Only run tests **not** matching this regular expression. The opposite of `--grep`. The filter does not apply to the tests from dependency projects, i.e. Playwright will still run all tests from project dependencies.
`--headed`	Run tests in headed browsers. Useful for debugging.

Option	Description
`--ignore-snapshots`	Whether to ignore snapshots. Use this when snapshot expectations are known to be different, e.g. running tests on Linux against Windows screenshots.
`--last-failed`	Only re-run the failures.
`--list`	list all the tests, but do not run them.
`--max-failures <N>` or `-x`	Stop after the first `N` test failures. Passing `-x` stops after the first failure.
`--no-deps`	Ignore the dependencies between projects and behave as if they were not specified.
`--output <dir>`	Directory for artifacts produced by tests, defaults to `test-results`.
`--only-changed [ref]`	Only run test files that have been changed between working tree and "ref". Defaults to running all uncommitted changes with ref=HEAD. Only supports Git.
`--pass-with-no-tests`	Allows the test suite to pass when no files are found.
`--project <name>`	Only run tests from the specified projects, supports '*' wildcard. Defaults to running all projects defined in the configuration file.
`--quiet`	Whether to suppress stdout and stderr from the tests.
`--repeat-each <N>`	Run each test `N` times, defaults to one.
`--reporter <reporter>`	Choose a reporter: minimalist `dot`, concise `line` or detailed `list`. See reporters for more information. You can also pass a path to a custom reporter file.
`--retries <number>`	The maximum number of retries for flaky tests, defaults to zero (no retries).
`--shard <shard>`	Shard tests and execute only selected shard, specified in the form `current/all`, 1-based, for example `3/5`.
`--timeout <number>`	Maximum timeout in milliseconds for each test, defaults to 30 seconds. Learn more about various timeouts.
`--trace <mode>`	Force tracing mode, can be `on`, `off`, `on-first-retry`, `on-all-retries`, `retain-on-failure`
`--tsconfig <path>`	Path to a single tsconfig applicable to all imported files. See tsconfig resolution for more details.
`--update-snapshots` or `-u`	Whether to update snapshots with actual results instead of comparing them. Use this when snapshot expectations have changed.
`--workers <number>` or `-j <number>`	The maximum number of concurrent worker processes that run in parallel.

Previous
Annotations

Next
Emulation