Playwright Browser test runner

CONCEPTS

Test filters are regexes on the file path, so "login" runs every file with login in its path.

Config projects, browsers
↓ `test`
Workers parallel files
↓ `reporter`
Report HTML, traces
↓ `show-trace`
Trace viewer timeline, DOM

Each project in the config is one browser or device setup. Select it with --project.

By default the config records a trace only on the first retry. Use --trace on to force it locally.

SETUP

`npm init playwright@latest`
Scaffold config, example tests and CI file

`npx playwright install`
Install the default browsers

`npx playwright install --with-deps`
Install browsers and system libraries

RUNNING TESTS

`npx playwright test`
Run every test headless

`npx playwright test <file>`
Run files matching a path regex

`npx playwright test <file>:<line>`
Run the test at a line

`npx playwright test -g "<title>"`
Run tests whose title matches

`npx playwright test --project=<name>`
Run one project (browser)

`npx playwright test --headed`
Show the browser window

`npx playwright test --last-failed`
Re-run only the failures

`npx playwright test --list`
List tests without running

DEBUGGING

`npx playwright test --ui`
Interactive UI mode with time travel

`npx playwright test --debug`
Run with the Inspector, headed, 1 worker

`npx playwright test --trace on`
Record a trace for every test

`npx playwright test --trace retain-on-failure`
Keep traces only for failures

`npx playwright show-trace <trace.zip>`
Open a trace in the viewer

PARALLELISM AND CI

`npx playwright test --workers <N>`
Number of workers or a percent of cores

`npx playwright test --workers=1`
Run serially in one worker

`npx playwright test --fully-parallel`
Run tests of one file in parallel too

`npx playwright test --retries <N>`
Retry failed tests N times

`npx playwright test --shard <n>/<total>`
Run one slice, e.g. 2/4

`npx playwright test --forbid-only`
Fail if test.only is left in

`npx playwright merge-reports <dir>`
Merge sharded blob reports

REPORTS AND OUTPUT

`npx playwright show-report`
Open the last HTML report

`npx playwright show-report <dir>`
Open a report folder or zip

`npx playwright test --reporter <name>`
Choose reporters: list, line, dot, html, json, junit

REPORTS AND OUTPUT (cont.)

`npx playwright test --reporter=html,list`
Combine reporters, comma separated

`npx playwright test --output <dir>`
Folder for artifacts (test-results)

SNAPSHOTS

`npx playwright test -u`
Update snapshots that changed

`npx playwright test -u all`
Rewrite every snapshot

`npx playwright test --update-snapshots missing`
Only write absent snapshots

CODEGEN

`npx playwright codegen <url>`
Record actions into test code

`npx playwright codegen -o <file>`
Save the generated script

`npx playwright codegen --target <lang>`
Pick the language, e.g. python

`npx playwright codegen -b <browser>`
Record in firefox or webkit

`npx playwright codegen --device "<name>"`
Emulate a device

`npx playwright codegen --save-storage <file>`
Save logged-in state

`npx playwright codegen --load-storage <file>`
Start from saved state

RECIPES

Debug one failing test

```
# run the test at line 42 with the Inspector
npx playwright test tests/login.spec.ts:42 --debug
```

Reuse a login

```
# log in once and save the state
npx playwright codegen --save-storage=auth.json <url>
# later sessions start logged in
npx playwright codegen --load-storage=auth.json <url>
```