



Tracing

API for collecting and saving Playwright traces. Playwright traces can be opened in [Trace Viewer](#) after Playwright script runs.

Start recording a trace before performing actions. At the end, stop tracing and save it to a file.

Sync**Async**

```
browser = chromium.launch()  
context = browser.new_context()  
context.tracing.start(screenshots=True, snapshots=True)  
page = context.new_page()  
page.goto("https://playwright.dev")  
context.tracing.stop(path = "trace.zip")
```

Methods

start

Added in: v1.12

Start tracing.

Usage

Sync**Async**

```
context.tracing.start(name="trace", screenshots=True, snapshots=True)  
page = context.new_page()  
page.goto("https://playwright.dev")  
context.tracing.stop(path = "trace.zip")
```

Arguments

- `name` `str` (*optional*)

If specified, the trace is going to be saved into the file with the given name inside the `traces_dir` folder specified in `browser_type.launch()`.

- `screenshots` `bool` (*optional*)

Whether to capture screenshots during tracing. Screenshots are used to build a timeline preview.

- `snapshots` `bool` (*optional*)

If this option is true tracing will

- capture DOM snapshot on every action
- record network activity

- `sources` `bool` (*optional*) Added in: v1.17

Whether to include source files for trace actions.

- `title` `str` (*optional*) Added in: v1.17

Trace name to be shown in the Trace Viewer.

start_chunk

Added in: v1.15

Start a new trace chunk. If you'd like to record multiple traces on the same `BrowserContext`, use `tracing.start()` once, and then create multiple trace chunks with `tracing.start_chunk()` and `tracing.stop_chunk()`.

Usage

Sync **Async**

```
context.tracing.start(name="trace", screenshots=True, snapshots=True)
page = context.new_page()
page.goto("https://playwright.dev")
```

```
context.tracing.start_chunk()  
page.get_by_text("Get Started").click()  
# Everything between start_chunk and stop_chunk will be recorded in the trace.  
context.tracing.stop_chunk(path = "trace1.zip")  
  
context.tracing.start_chunk()  
page.goto("http://example.com")  
# Save a second trace file with different actions.  
context.tracing.stop_chunk(path = "trace2.zip")
```

Arguments

- `name` `str` (*optional*) Added in: v1.32

If specified, the trace is going to be saved into the file with the given name inside the `traces_dir` folder specified in `browser_type.launch()`.

- `title` `str` (*optional*) Added in: v1.17

Trace name to be shown in the Trace Viewer.

stop

Added in: v1.12

Stop tracing.

Usage

```
tracing.stop()  
tracing.stop(**kwargs)
```

Arguments

- `path` `Union[str, pathlib.Path]` (*optional*)

Export trace into the file with the given path.

stop_chunk

Added in: v1.15

Stop the trace chunk. See [tracing.start_chunk\(\)](#) for more details about multiple trace chunks.

Usage

```
tracing.stop_chunk()  
tracing.stop_chunk(**kwargs)
```

Arguments

- `path` [Union\[str, pathlib.Path\]](#) (*optional*)

Export trace collected since the last [tracing.start_chunk\(\)](#) call into the file with the given path.