Trace viewer

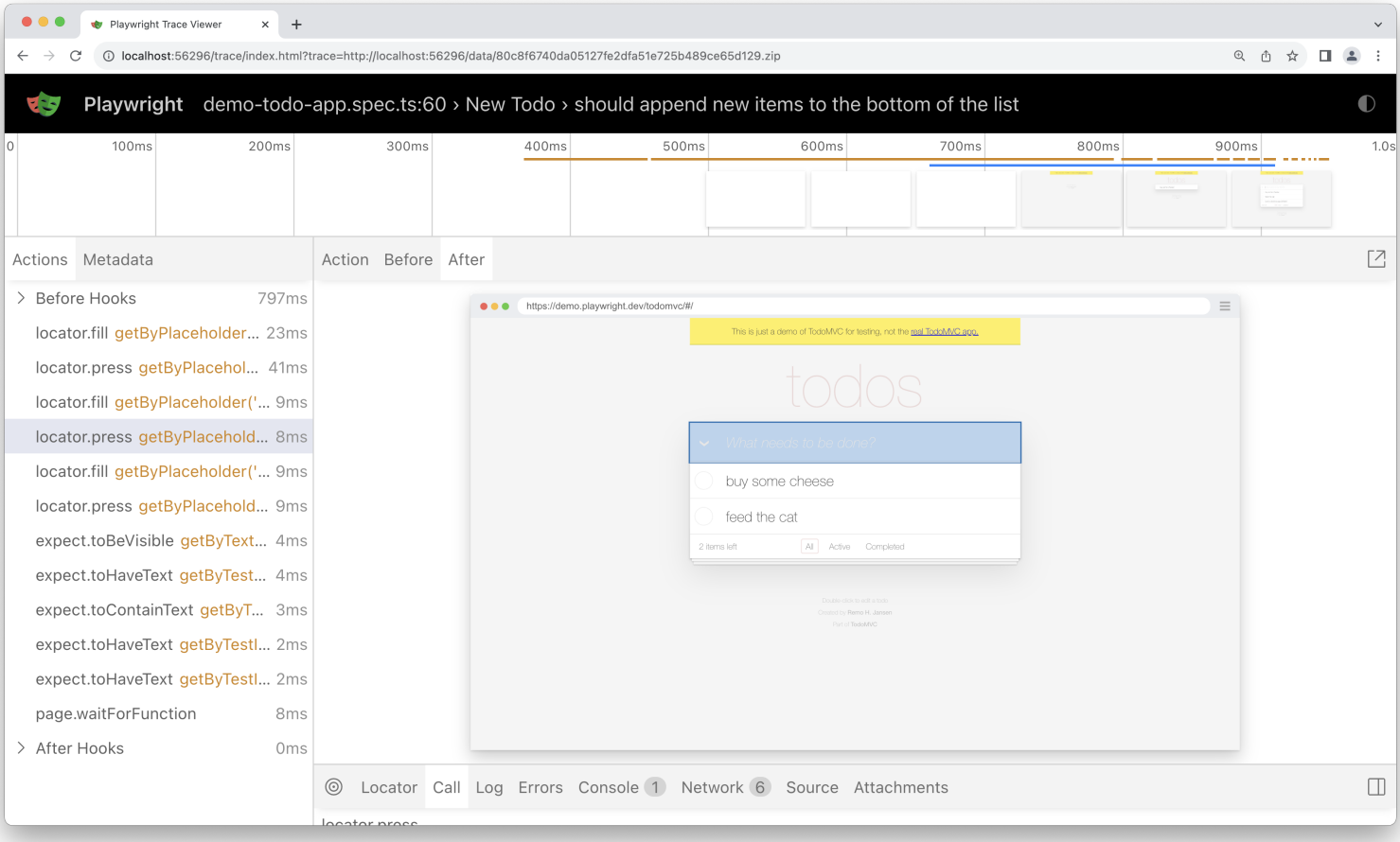
Introduction

Playwright Trace Viewer is a GUI tool that helps you explore recorded Playwright traces after the script has ran. Traces are a great way for debugging your tests when they fail on CI. You can open traces locally or in your browser on trace.playwright.dev.

Trace Viewer features

Actions

In the Actions tab you can see what locator was used for every action and how long each one took to run. Hover over each action of your test and visually see the change in the DOM snapshot. Go back and forward in time and click an action to inspect and debug. Use the Before and After tabs to visually see what happened before and after the action.



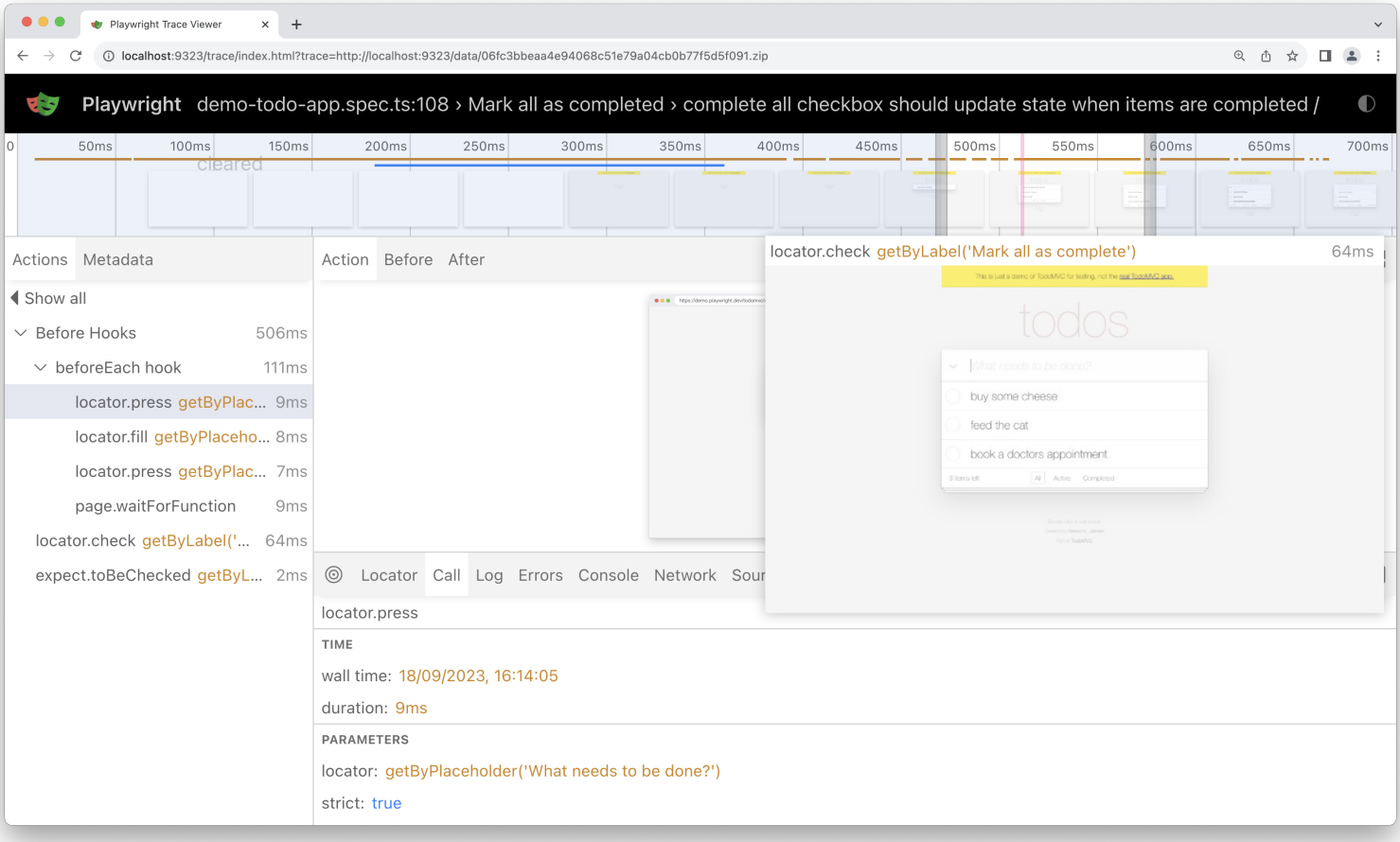
Selecting each action reveals:

- action snapshots
- action log
- source code location

Screenshots

When tracing with the `screenshots` option turned on, each trace records a screencast and renders it as a film strip. You can hover over the film strip to see a magnified image of for each action and state which helps you easily find the action you want to inspect.

Double click on an action to see the time range for that action. You can use the slider in the timeline to increase the actions selected and these will be shown in the Actions tab and all console logs and network logs will be filtered to only show the logs for the actions selected.

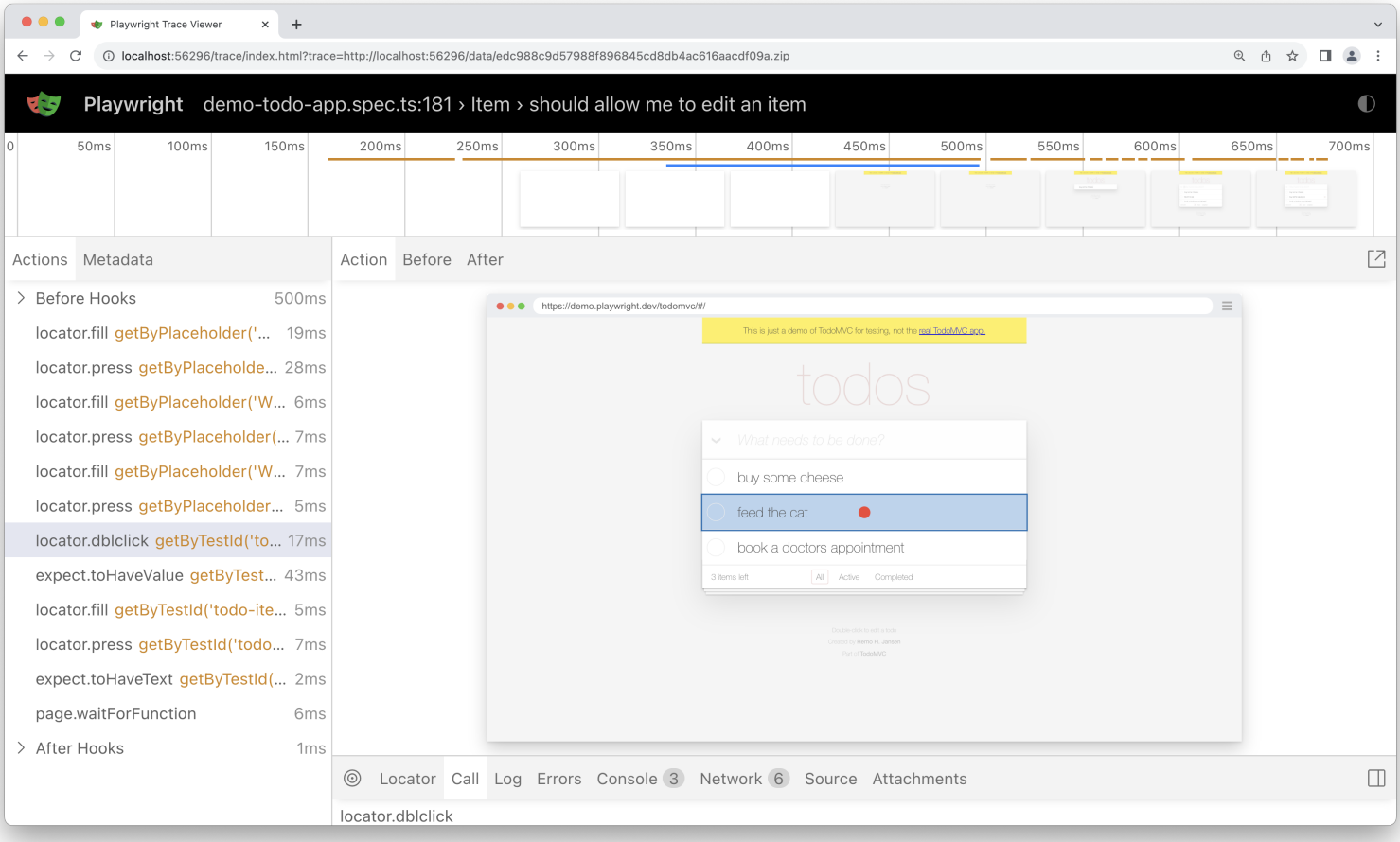


Snapshots

When tracing with the `snapshots` option turned on (default), Playwright captures a set of complete DOM snapshots for each action. Depending on the type of the action, it will capture:

Type	Description
Before	A snapshot at the time action is called.
Action	A snapshot at the moment of the performed input. This type of snapshot is especially useful when exploring where exactly Playwright clicked.
After	A snapshot after the action.

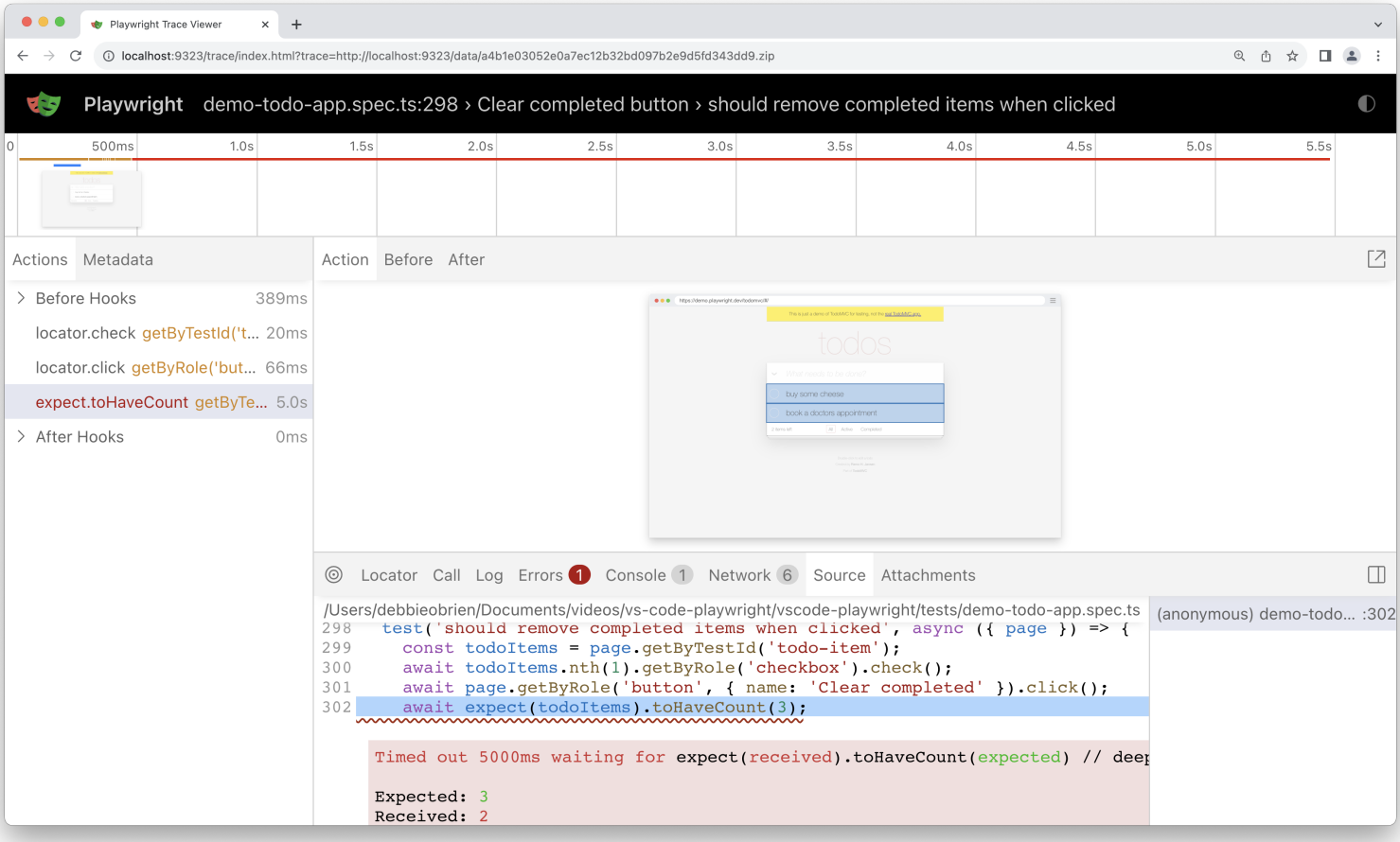
Here is what the typical Action snapshot looks like:



Notice how it highlights both, the DOM Node as well as the exact click position.

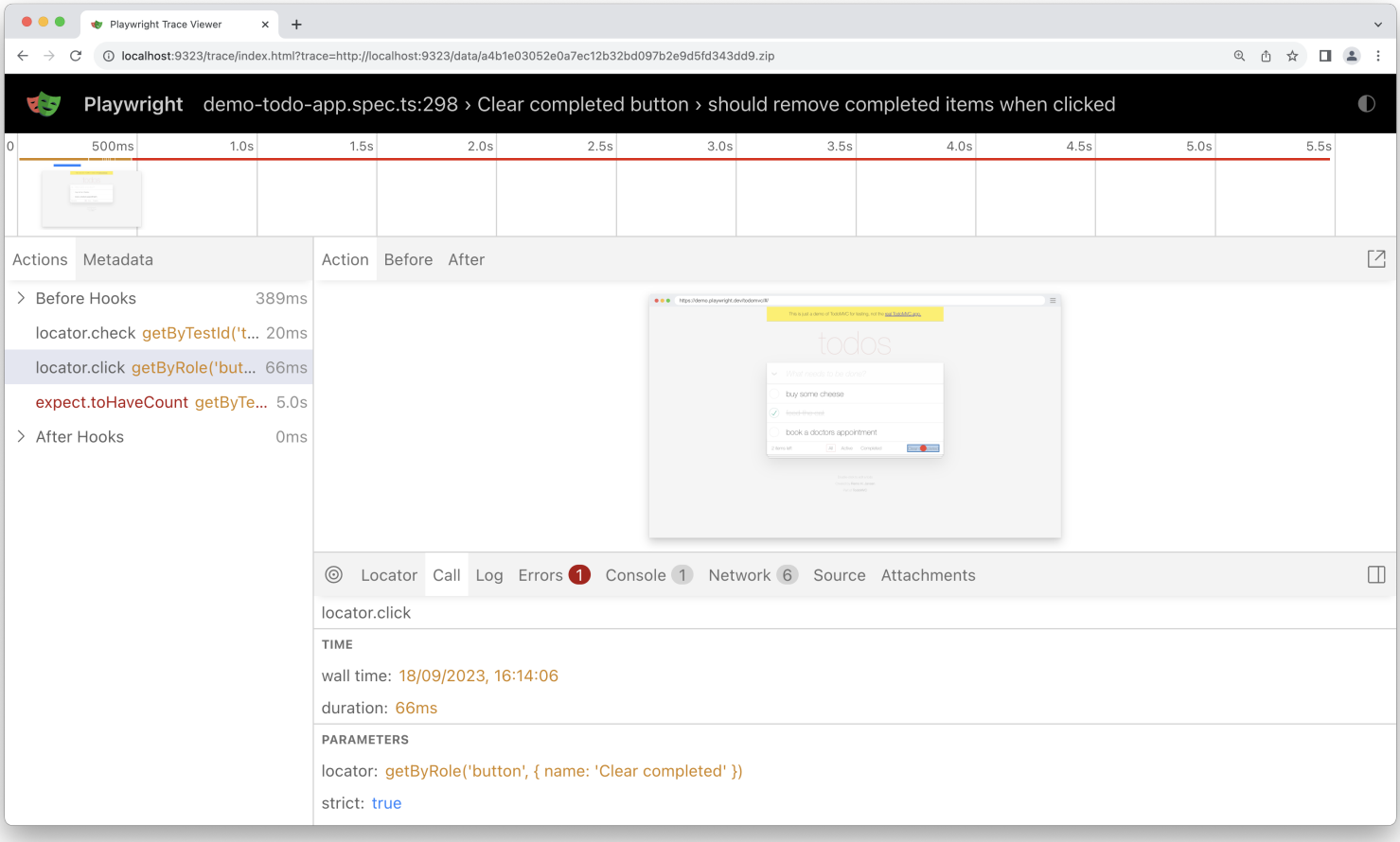
Source

As you hover over each action of your test the line of code for that action is highlighted in the source panel.



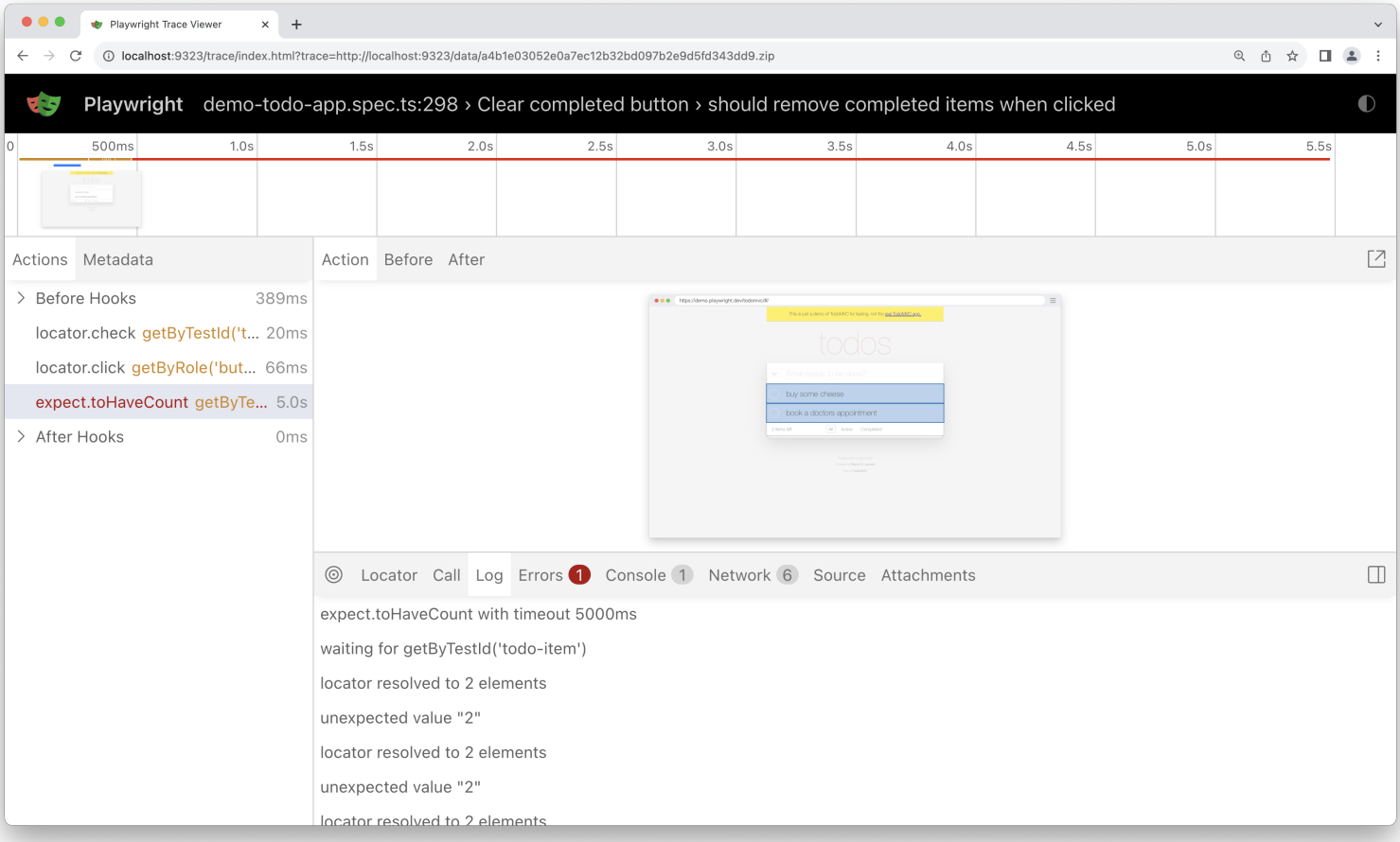
Call

The call tab shows you information about the action such as the time it took, what locator was used, if in strict mode and what key was used.



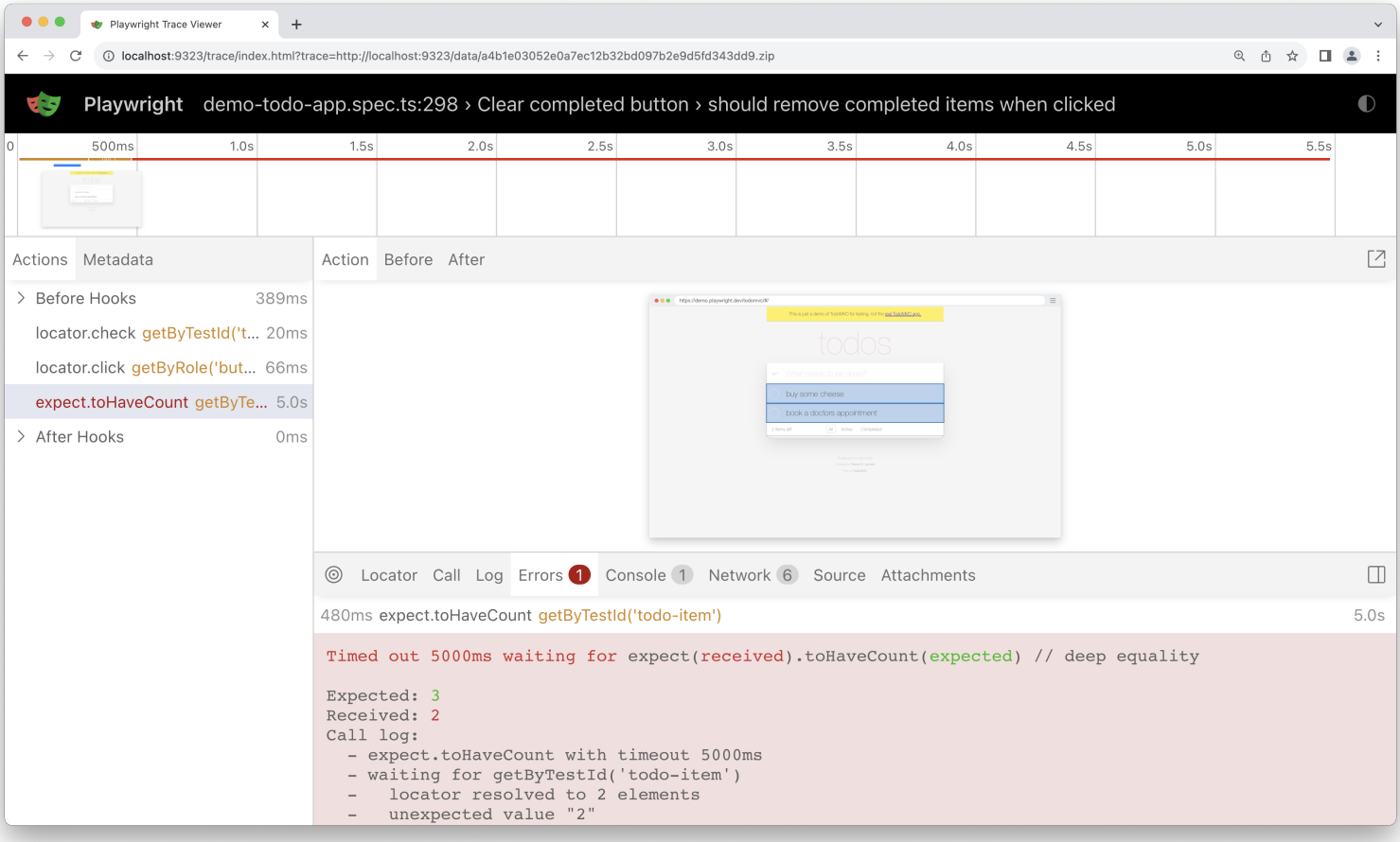
Log

See a full log of your test to better understand what Playwright is doing behind the scenes such as scrolling into view, waiting for element to be visible, enabled and stable and performing actions such as click, fill, press etc.



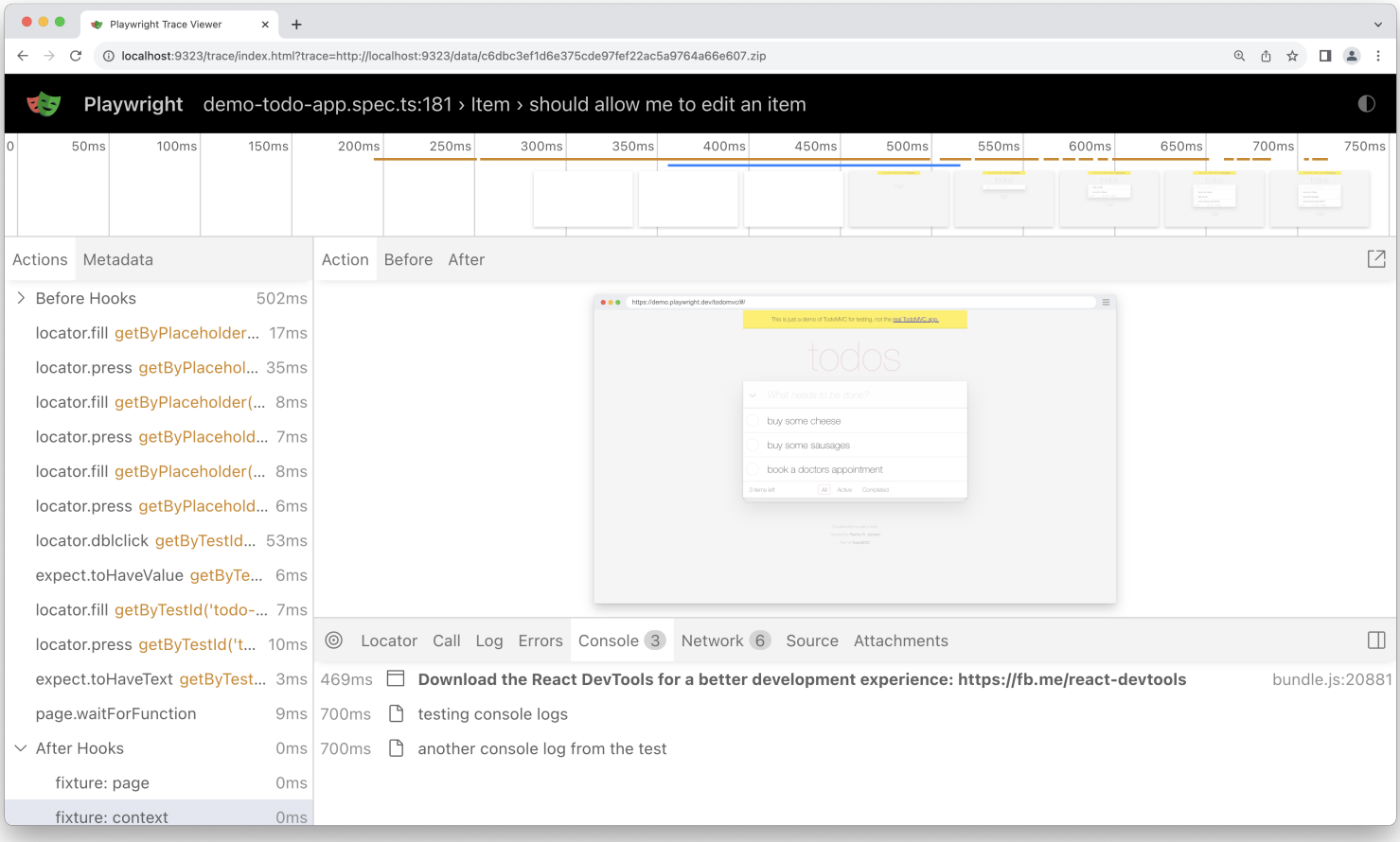
Errors

If your test fails you will see the error messages for each test in the Errors tab. The timeline will also show a red line highlighting where the error occurred. You can also click on the source tab to see on which line of the source code the error is.



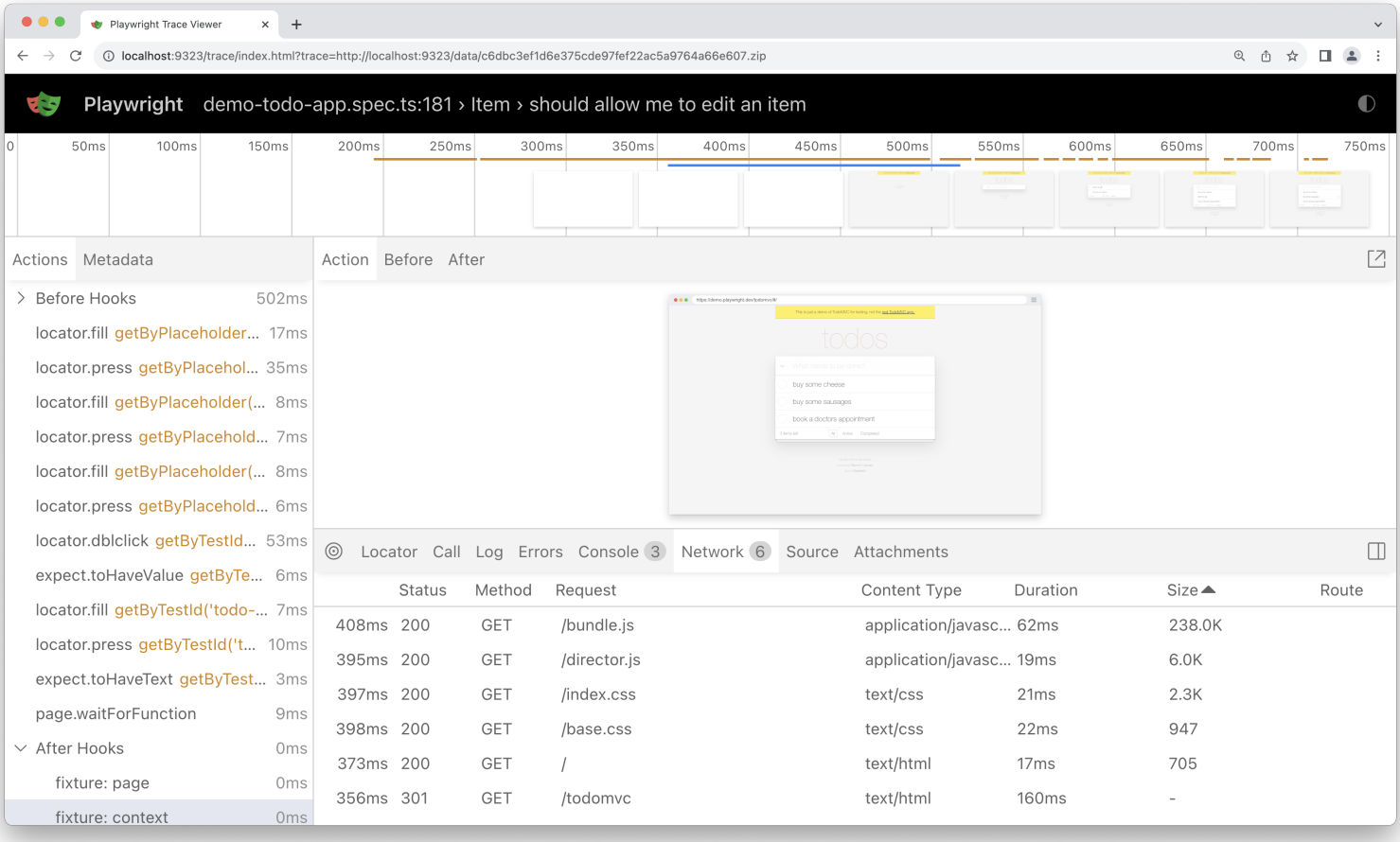
Console

See console logs from the browser as well as from your test. Different icons are displayed to show you if the console log came from the browser or from the test file.



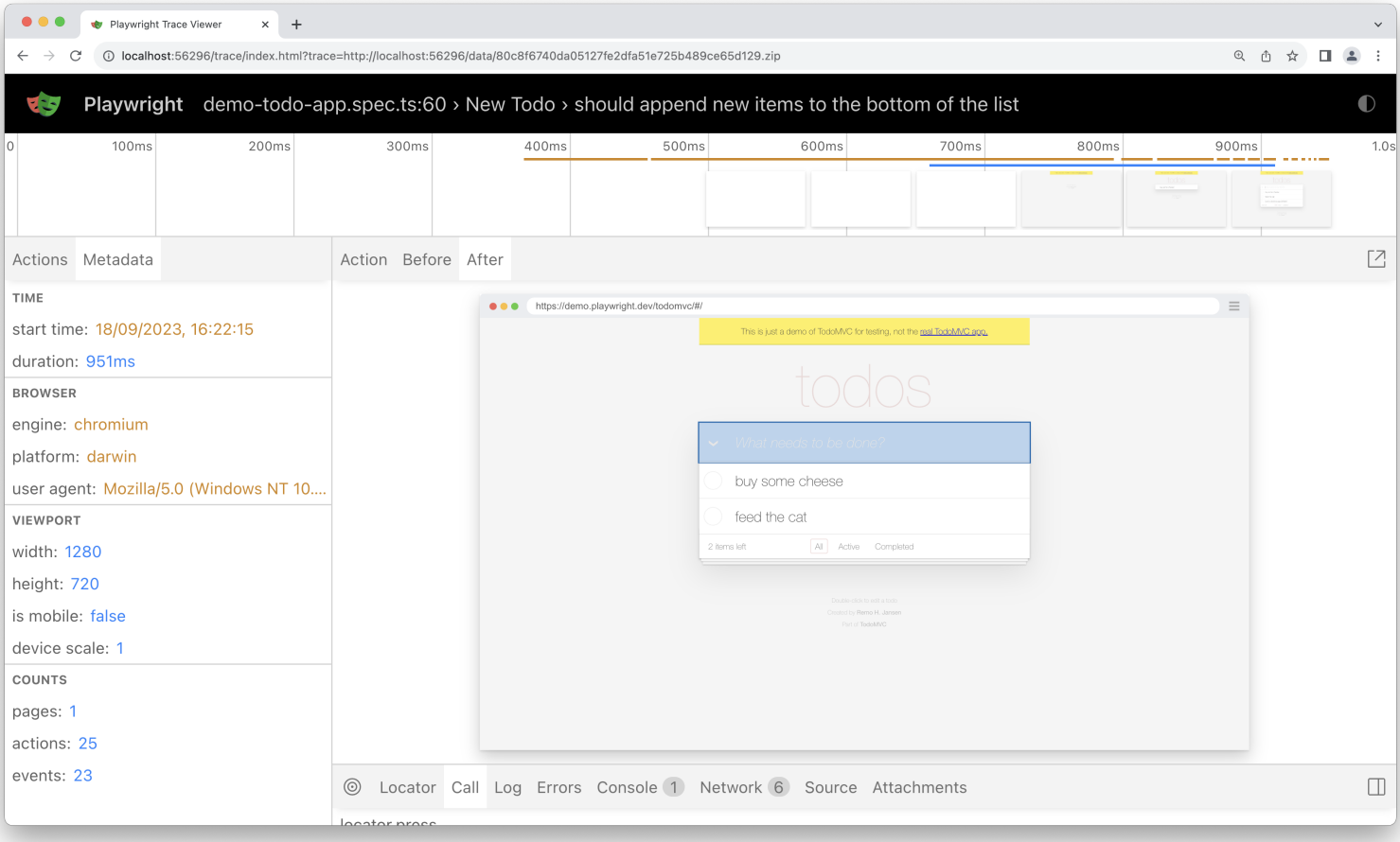
Network

The Network tab shows you all the network requests that were made during your test. You can sort by different types of requests, status code, method, request, content type, duration and size. Click on a request to see more information about it such as the request headers, response headers, request body and response body.



Metadata

Next to the Actions tab you will find the Metadata tab which will show you more information on your test such as the Browser, viewport size, test duration and more.



Recording a trace

Traces can be recorded by running your tests with the `--tracing` flag.

```
pytest --tracing on
```

Options for tracing are:

- `on`: Record trace for each test
- `off`: Do not record trace. (default)
- `retain-on-failure`: Record trace for each test, but remove all traces from successful test runs.

This will record the trace and place it into the file named `trace.zip` in your `test-results` directory.

▸ If you are not using Pytest, click here to learn how to record traces.

Opening the trace

You can open the saved trace using the Playwright CLI or in your browser on `trace.playwright.dev`. Make sure to add the full path to where your `trace.zip` file is located. This should include the full path to your `trace.zip` file.

```
playwright show-trace trace.zip
```

Using trace.playwright.dev

trace.playwright.dev is a statically hosted variant of the Trace Viewer. You can upload trace files using drag and drop.

Viewing remote traces

You can open remote traces using it's URL. They could be generated on a CI run which makes it easy to view the remote trace without having to manually download the file.

```
playwright show-trace https://example.com/trace.zip
```

You can also pass the URL of your uploaded trace (e.g. inside your CI) from some accessible storage as a parameter. CORS (Cross-Origin Resource Sharing) rules might apply.

```
https://trace.playwright.dev/?
trace=https://demo.playwright.dev/reports/todomvc/data/cb0fa77ebd9487a5c899f3ae65a7f
```