

[Skip to main content](#)



[PlaywrightDocsAPI](#)

[Node.js](#)

- [Node.js](#)
- [Python](#)
- [Java](#)
- [.NET](#)

[Community](#)

Search⌕

- [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
- Web server

On this page

# Web server

## Introduction

Playwright comes with a `webserver` option in the config file which gives you the ability to launch a local dev server before running your tests. This is ideal for when writing your tests during development and when you don't have a staging or production url to test against.

## Configuring a web server

Use the `webserver` property in your Playwright config to launch a development web server during the tests.

```
playwright.config.ts
import { defineConfig } from '@playwright/test';

export default defineConfig({
  // Run your local dev server before starting the tests
  webServer: {
    command: 'npm run start',
    url: 'http://127.0.0.1:3000',
    reuseExistingServer: !process.env.CI,
    stdout: 'ignore',
    stderr: 'pipe',
  },
});
```

Property	Description
<a href="#">testConfig.webServer</a>	Launch a development web server (or multiple) during the tests.
command	Shell command to start the local dev server of your app.
url	URL of your http server that is expected to return a 2xx, 3xx, 400, 401, 402, or 403 status code when the server is ready to accept connections.
reuseExistingServer	If <code>true</code> , it will re-use an existing server on the url when available. If no server is running on that url, it will run the command to start a new server. If <code>false</code> , it will throw if an existing process is listening on the url. To see the stdout, you can set the <code>DEBUG=pw:webserver</code> environment variable.
ignoreHTTPSErrors	Whether to ignore HTTPS errors when fetching the url. Defaults to <code>false</code> .

Property	Description
cwd	Current working directory of the spawned process, defaults to the directory of the configuration file.
stdout	If "pipe", it will pipe the stdout of the command to the process stdout. If "ignore", it will ignore the stdout of the command. Default to "ignore".
stderr	Whether to pipe the stderr of the command to the process stderr or ignore it. Defaults to "pipe".
timeout	How long to wait for the process to start up and be available in milliseconds. Defaults to 60000.

## Adding a server timeout

Webservers can sometimes take longer to boot up. In this case, you can increase the timeout to wait for the server to start.

```
playwright.config.ts
import { defineConfig } from '@playwright/test';

export default defineConfig({
  // Rest of your config...

  // Run your local dev server before starting the tests
  webServer: {
    command: 'npm run start',
    url: 'http://127.0.0.1:3000',
    reuseExistingServer: !process.env.CI,
    timeout: 120 * 1000,
  },
});
```

## Adding a baseURL

It is also recommended to specify the `baseURL` in the `use: {}` section of your config, so that tests can use relative urls and you don't have to specify the full URL over and over again.

When using [page.goto\(\)](#), [page.route\(\)](#), [page.waitForURL\(\)](#), [page.waitForRequest\(\)](#), or [page.waitForResponse\(\)](#) it takes the base URL in consideration by using the [URL\(\)](#) constructor for building the corresponding URL. For Example, by setting the `baseURL` to `http://127.0.0.1:3000` and navigating to `/login` in your tests, Playwright will run the test using `http://127.0.0.1:3000/login`.

```
playwright.config.ts
import { defineConfig } from '@playwright/test';

export default defineConfig({
  // Rest of your config...

  // Run your local dev server before starting the tests
  webServer: {
    command: 'npm run start',
    url: 'http://127.0.0.1:3000',
```

```

    reuseExistingServer: !process.env.CI,
  },
  use: {
    baseUrl: 'http://127.0.0.1:3000',
  },
});

```

Now you can use a relative path when navigating the page:

test.spec.ts

```

import { test } from '@playwright/test';

test('test', async ({ page }) => {
  // This will navigate to http://127.0.0.1:3000/login
  await page.goto('./login');
});

```

## Multiple web servers

Multiple web servers (or background processes) can be launched simultaneously by providing an array of `webServer` configurations. See [testConfig.webServer](#) for more info.

playwright.config.ts

```

import { defineConfig } from '@playwright/test';

export default defineConfig({
  webServer: [
    {
      command: 'npm run start',
      url: 'http://127.0.0.1:3000',
      timeout: 120 * 1000,
      reuseExistingServer: !process.env.CI,
    },
    {
      command: 'npm run backend',
      url: 'http://127.0.0.1:3333',
      timeout: 120 * 1000,
      reuseExistingServer: !process.env.CI,
    }
  ],
  use: {
    baseUrl: 'http://127.0.0.1:3000',
  },
});

```

[Previous](#)  
[UI Mode](#)

[Next](#)  
[Library](#)

- [Introduction](#)
- [Configuring a web server](#)
- [Adding a server timeout](#)

- [Adding a baseURL](#)
- [Multiple web servers](#)

## Learn

- [Getting started](#)
- [Playwright Training](#)
- [Learn Videos](#)
- [Feature Videos](#)

## Community

- [Stack Overflow](#)
- [Discord](#)
- [Twitter](#)
- [LinkedIn](#)

## More

- [GitHub](#)
- [YouTube](#)
- [Blog](#)
- [Ambassadors](#)

Copyright © 2024 Microsoft