

[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](#)

- [Migrating from Protractor](#)
 - [Migrating from Puppeteer](#)
 - [Migrating from Testing Library](#)
- [Integrations](#)
 - [Docker](#)
 - [Continuous Integration](#)
 - [Selenium Grid \(experimental\)](#)
- [Supported languages](#)
-
- Supported languages

On this page

Supported languages

Introduction

Playwright is available in multiple languages that share the same underlying implementation. All core features for automating the browser are supported in all languages, while testing ecosystem integration is different. Pick the language based on your experience, familiarity with its testing ecosystem and your project constraints. For the best experience pick the test runner that we recommend for each language.

JavaScript and TypeScript

Playwright for Node.js comes with its own [test runner](#) that provides great parallelization mechanism, screenshot assertions, html reporter, automatic tracing etc.

- [Documentation](#)
- [GitHub repo](#)

Python

Playwright [Pytest plugin](#) is the recommended way to run end-to-end tests. It provides context isolation, running it on multiple browser configurations and more out of the box.

- [Documentation](#)
- [GitHub repo](#)

Java

You can choose any testing framework such as JUnit or TestNG based on your project requirements.

- [Documentation](#)
- [GitHub repo](#)

.NET

Playwright for .NET comes with [MSTest base classes](#) and [NUnit base classes](#) for writing end-to-end tests.

- [Documentation](#)
- [GitHub repo](#)

[Previous](#)

[Selenium Grid \(experimental\)](#)

- [Introduction](#)
- [JavaScript and TypeScript](#)
- [Python](#)
- [Java](#)
- [.NET](#)

Learn

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

Community

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

More

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

Copyright © 2024 Microsoft