

Automation Testing with Playwright and AI

Build automation that is readable, stable and useful to a real delivery team.

LEVEL

Beginner to advanced

FORMAT

Live online

PRICE

€199 one-time

Explore the course and enrol



SCAN TO EXPLORE

THE PROMISE

Build automation that is readable, stable and useful to a real delivery team.

Learn JavaScript for testing, automate browser and application programming interface checks with Playwright, and use AI as a careful testing assistant.

WHO THIS IS FOR

Manual testers, junior automation testers and developers who want a structured route into modern test automation.

STARTING POINT

Basic testing knowledge is helpful. JavaScript is taught from the beginning.

HOW YOU LEARN

Live interactive sessions, guided assignments and practical work.

ONGOING SUPPORT

WhatsApp group, recordings, study material and continued access after the course.

COURSE EXPERIENCE

Learn live. Practise consistently. Build work you can explain.

WHAT YOU WILL BE ABLE TO DO

Skills you can use, show and explain.

01

Write clear JavaScript for automated tests

02

Automate reliable browser journeys with Playwright

03

Choose stable locators and useful assertions

04

Build page objects, helpers and reusable test data

05

Test APIs and combine API setup with user interface checks

06

Use AI to explore, review and debug without trusting it blindly

MODULE-BY-MODULE SYLLABUS

A clear path from foundation to practical result.

M01

Introduction to test automation

Understand the automation landscape and why Playwright is a strong modern choice.

- Software development and testing life cycles
- Manual and automation testing
- Testing types and automation suitability
- Playwright, Selenium and Cypress comparison
- Playwright architecture and supported browsers
- Course roadmap

M02

JavaScript fundamentals

Build the programming foundation needed for readable and reliable automation.

- Variables, data types and operators
- Conditions, loops and functions
- Arrays, objects and string methods
- JSON and JavaScript modules
- Arrow functions, destructuring and template literals
- Promises, async and await, and error handling

M03

TypeScript essentials

Use TypeScript to make test code safer, clearer and easier to maintain.

- Types, interfaces and classes
- Enums and generics
- Optional parameters and type assertions
- TypeScript with Playwright

MODULES 04 TO 22

M04

Environment setup

Create a dependable local workflow for Playwright development.

- Node.js, Visual Studio Code and Git
- NPM commands
- Playwright and browser installation
- Project structure and Playwright configuration

M05

Playwright basics

Write and run tests confidently with the Playwright test runner.

- First test and test runner
- Fixtures and hooks
- Parallel execution and projects
- Tags, annotations and command-line tools

M06

Locators

Find elements using resilient selectors that reflect how people use the interface.

- ID, CSS, XPath and text locators
- Role, label and placeholder locators
- Alternative text and test ID locators
- Locator chaining and best practices

M07

Web user interface automation

Automate common browser interactions and dynamic user journeys.

- Clicking, filling and form controls
- Dropdowns and auto-suggestions
- Mouse and keyboard actions
- File uploads and downloads
- Date pickers, tables and dynamic elements

MODULES 08 TO 22

M08

Advanced user interface automation

Handle complex browser behaviour without creating fragile tests.

- Frames and nested frames
- Multiple windows and tabs
- Alerts and authentication pop-ups
- Shadow DOM and drag and drop
- Infinite scroll and canvas basics

M09

Waits and synchronisation

Make tests stable by waiting for meaningful application states.

- Playwright auto-waiting
- Explicit waits and selector states
- Load state, response and URL waits
- Timeout management

M10

Assertions

Prove expected behaviour with focused and readable checks.

- Built-in assertions
- Soft and custom assertions
- API assertions
- Negative assertions

M11

Page Object Model

Separate page behaviour from test intent using a maintainable design.

- Framework design and base pages
- Page classes
- Utilities and reusable components
- Page Object Model best practices

MODULES 12 TO 22

M12

Data-driven testing

Run useful test variations with structured, reusable data.

- JSON, CSV and Excel data
- Environment variables
- Faker library
- Parameterized tests

M13

API testing

Validate backend behaviour and connect API setup with browser flows.

- REST API fundamentals
- GET, POST, PUT, PATCH and DELETE
- Headers and authentication
- Request context and response validation
- Chaining API calls

M14

Reporting

Create evidence that helps the team understand results and failures.

- HTML reports and Trace Viewer
- Screenshots and videos
- Allure reports
- Custom reporting

M15

Debugging

Diagnose failures quickly using Playwright and editor tools.

- Playwright Inspector
- Visual Studio Code debugger
- Trace Viewer and slow motion
- Console logging

MODULES 16 TO 22

M16

Cross-browser testing

Run meaningful coverage across browsers and representative devices.

- Chromium, Firefox and WebKit
- Mobile emulation
- Device testing
- Cross-browser project configuration

M17

Framework development

Bring configuration, utilities and reliability patterns into one framework.

- Folder structure and configuration management
- Constants, utilities and helpers
- Logging
- Retry logic

M18

Continuous integration and delivery

Run automation consistently in shared delivery pipelines.

- Git and GitHub workflows
- GitHub Actions
- Jenkins pipelines
- Pipeline and scheduled execution

M19

Docker basics

Package the execution environment for consistent automation runs.

- Docker installation
- Images and containers
- Running Playwright in Docker
- Repeatable local and pipeline execution

MODULES 20 TO 22

M20

AI-powered automation

Use AI to accelerate automation work while retaining human verification.

- GitHub Copilot
- Prompt engineering for test cases
- AI-assisted locator and script generation
- Model Context Protocol
- AI-assisted debugging and review

M21

Real-world automation project

Build a complete automation framework around realistic product behaviour.

- JavaScript and Playwright framework
- Page Object Model and API testing
- Allure reporting
- GitHub Actions and Jenkins
- Environment configuration

M22

Interview preparation

Prepare to explain your technical decisions and automation experience clearly.

- Playwright and JavaScript questions
- API testing and framework design questions
- Coding challenges
- CV preparation and mock interviews

NEXT / PRACTICAL PROJECT

Production-style Playwright suite

Build a focused automation project around realistic user journeys. Your final suite will combine browser tests, API support, reusable design and clear reporting.

BRING THE LEARNING TOGETHER

Production-style Playwright suite

Build a focused automation project around realistic user journeys. Your final suite will combine browser tests, API support, reusable design and clear reporting.

OUTPUT 01

Automation strategy and scope

OUTPUT 02

Playwright framework structure

OUTPUT 03

User interface and API tests

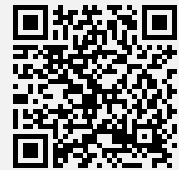
OUTPUT 04

Execution report and technical walkthrough

SUPPORT THAT CONTINUES

Recordings and practice material stay available in Google Drive. The course WhatsApp group remains open for course-related questions after completion.

Ready to make this skill real?



Review the course page, confirm any questions and enrol securely when you are ready.

SCAN TO OPEN COURSE

COURSE ENROLMENT

Playwright and AI Automation

€199 one-time payment

Explore course and enrol securely

01

Live online sessions

01

Daily and weekly assignments

01

Recordings and study material

01

Mentor and coordinator support

Ask a question on WhatsApp