Web-Testautomatisierung mit Playwright

Moderne Webanwendungen mit Playwright zuverlässig automatisiert testen

01 INTRO

Playwright verbindet Browserautomatisierung mit Funktionen, die speziell für moderne Webanwendungen und stabile automatisierte Tests entwickelt wurden. Automatische Synchronisation, browserübergreifende Ausführung, Netzwerkzugriff, Tracing und isolierte Browser-Kontexte reduzieren viele klassische Probleme der UI-Testautomatisierung.

Der Kurs vermittelt Playwright von den Grundlagen bis zum Aufbau einer wartbaren Web-Testlösung. Neben Browserinteraktion stehen robuste Selektoren, Synchronisation, Teststruktur, Daten, Debugging, Parallelisierung und CI/CD im Mittelpunkt.

DAUER

3 Tage

FORMAT

Präsenz · Live Online · Inhouse

VORKENNTNISSE

Grundkenntnisse in Testautomatisierung und Programmierung

02 TRAININGSPROGRAMM

Inhalte

Playwright und moderne Browserautomatisierung

- Architektur
- Browser
- Browser Contexts
- Pages
- Playwright Test
- Auto-Waiting
- Isolation
- Einsatzbereiche

Projekt einrichten

- Installation
- Konfiguration
- Teststruktur
- Browserprojekte
- lokale Ausführung
- Entwicklungsumgebung
- erste Tests

Elemente zuverlässig finden

- Locators
- Rollen
- Labels
- Text
- Test IDs
- CSS und XPath
- Locator-Strategie
- robuste versus fragile Selektoren

Benutzerinteraktionen

- Click
- Fill
- Select
- Checkboxen
- Tastatur
- Maus
- Upload und Download
- Dialoge

Assertions und Synchronisation

- Web Assertions
- Auto-Retry
- Sichtbarkeit
- Inhalte
- URLs
- Zustände
- explizite Waits
- unnötige Sleeps vermeiden

Navigation und Browserkontexte

- Navigation
- Tabs und Popups
- Frames
- Cookies
- Sessions
- Authentication State
- isolierte Kontexte

Teststruktur und Fixtures

- Test Suites
- Hooks
- Fixtures
- eigene Fixtures
- Setup und Teardown
- Wiederverwendung
- Testisolation

Wartbare Testarchitektur

- Page Objects
- Component Objects
- fachliche Aktionen
- Hilfsschichten
- Assertions
- geeignete Abstraktion
- Testcode strukturieren

Netzwerk und APIs

- Requests beobachten
- Responses untersuchen
- Routing
- Mocking
- API Requests
- Backend-Zustände vorbereiten
- UI- und API-Tests kombinieren

Debugging und Diagnose

- Inspector
- Debug Mode
- Trace Viewer
- Screenshots
- Videos
- Netzwerkdaten
- Fehlerkontext
- Flaky Tests analysieren

Parallelisierung und CI/CD

- Worker
- Parallel Tests
- Sharding
- Retries
- Browser Matrix
- Reports
- Pipeline-Ausführung
- Artefakte

Praktisches Playwright-Projekt

- Webanwendung analysieren
- Locator-Strategie entwickeln
- Tests implementieren
- Architektur strukturieren
- Daten und Sessions behandeln
- Fehler diagnostizieren
- Tests in CI ausführen

03 DURCHFÜHRUNG

Rahmen und Organisation

Dauer	3 Tage
Durchführung	Präsenz oder Live Online, vorzugsweise als Inhouse-Schulung
Schwerpunkt	Inhouse-Schulungen für Unternehmen und Teams
Sprache	Deutsch, englische Fachbegriffe und Dokumentation
Unterlagen	Kursunterlagen, Beispiele und Übungsaufgaben
Technik	Eigener Rechner mit Node.js oder Python und installierten Browsern

> **Inhouse-Schulungen**
> Die Übungen können mit TypeScript oder Python durchgeführt und an die eigene Webanwendung angepasst werden.

04 ZIELGRUPPE

Für wen ist der Kurs gedacht?

Tester, Testautomatisierer, Quality Engineers und Entwickler, die moderne Webanwendungen mit Playwright automatisiert testen möchten.

> **Voraussetzungen**
> Grundlegende Programmiererfahrung sowie Grundlagen des Softwaretests. Kenntnisse von HTML und Webtechnologien sind hilfreich.

05 LERNZIELE

Was Sie nach dem Kurs können

Die Teilnehmer können Playwright-Projekte aufsetzen, robuste Webtests entwickeln, Synchronisation und Isolation beherrschen, Tests strukturieren und in CI/CD-Umgebungen ausführen.

06 FAQ

Häufige Fragen

Wird TypeScript oder Python verwendet?

Der Kurs kann abhängig von Zielgruppe und Projektumfeld mit TypeScript oder Python durchgeführt werden. Die Playwright-Konzepte bleiben weitgehend gleich.

Was unterscheidet den Kurs vom Selenium-Kurs?

Dieser Kurs behandelt Playwright und dessen Architektur und Möglichkeiten. Der Selenium-Kurs konzentriert sich entsprechend auf Selenium und dessen Ökosystem.

Werden nur UI-Tests behandelt?

Browserautomatisierung bildet den Schwerpunkt. Playwright-Funktionen für API-Zugriffe und Netzwerksteuerung werden dort genutzt, wo sie robuste Webtests unterstützen.

UC IT Service · Ulrich Cuber · kontakt@uc-it.de
https://www.uc-it.de/coaching/playwright/