

Strona: awesome-testing.com

Kontakt: <https://awesome-testing.com/contact>

Playwright + LLM + MCP w testach API i UI

Warsztat łączący Playwright, MCP i narzędzia AI do przyspieszenia testów eksploracyjnych oraz automatycznych.

Poziom: zaawansowany

Czas trwania: 2 days

Tryb: zdalnie, stacjonarnie

Języki: PL, EN

Narzędzia: ChatGPT, Cursor, Copilot, Codex, Playwright, MCP, JavaScript, TypeScript, Playwright MCP, AI agents

Szczegóły: <https://awesome-testing.com/szkolenia/playwright-llm-mcp>

Najważniejsze efekty

- Connect AI tools with Playwright through MCP and tool calling
- Automate API and UI test flows with controlled AI assistance
- Evaluate productivity gains and reliability risks in AI-driven testing

Pełny program

Cel szkolenia

Szkolenie pokazuje, jak połączyć serwer Model Context Protocol (MCP) dla Playwrighta z narzędziami AI (np. GitHub Copilot / Cursor) oraz podejściami takimi jak tool calling (function calling) i Retrieval-Augmented Generation (RAG). Celem jest przyspieszenie testów eksploracyjnych oraz tworzenia i utrzymania testów automatycznych aplikacji webowych.

Po ukończeniu szkolenia uczestnik będzie potrafił:

- Wyjaśnić, czym są tool calling (function calling) i RAG oraz jak wykorzystać je w testowaniu.
- Opisać rolę MCP i jego architekturę w ekosystemie Playwright.
- Przeprowadzić sesję testów eksploracyjnych wspieraną przez LLM.
- Generować i utrzymywać testy Playwright (UI i API) z pomocą AI.
- Ocenić, kiedy AI realnie skraca czas pracy, a kiedy zwiększa ryzyko.

Zakres tematyczny

1) Nowoczesne techniki LLM

- Podstawy LLM (jak działają narzędzia typu ChatGPT)
- Tool calling / narzędzia: łączenie modelu z zewnętrznymi możliwościami
- Retrieval-Augmented Generation (RAG): poprawa jakości odpowiedzi dzięki dokumentacji
- AI agenci: koncepcje, teoria i zastosowania

2) MCP i Playwright

- Architektura serwera MCP
- Instalacja, konfiguracja i integracja z Playwright (JS/TS)
- Przegląd API: snapshoty dostępności, nawigacja i akcje na elementach

3) MCP w praktyce testowej

- Testy eksploracyjne sterowane promptami
- Generowanie przypadków testowych na podstawie user stories

4) Automatyzacja testów z AI

- Zastosowanie Copilota / Cursora w automatyzacji

- Połączenie Copilota / Cursora z MCP: workflow „prompt → kod → uruchom → napraw”
- Warsztaty praktyczne:
 - Automatyzacja testów REST API w Playwright
 - Automatyzacja testów UI z MCP + Playwright
- Analiza produktywności: co pisać samodzielnie, a co generować

5) Ryzyka, etyka i utrzymanie

- Halucynacje modeli a stabilność i jakość testów
- Wersjonowanie promptów i odtwarzalność sesji
- Analiza kosztów i opłacalności
- Roadmapa MCP i Playwright: co warto obserwować

Jak się przygotować

Dla kogo?

Szkolenie jest przeznaczone dla testerów znających podstawy programowania w JavaScript/TypeScript oraz podstawy Playwrighta. Mile widziana jest znajomość Gita.

Co przygotować

- Laptop z zainstalowanym Node.js (LTS) , Git i VS Code
- Skonfigurowane narzędzie AI (np. GitHub Copilot lub Cursor)

Szczegółową checklistę środowiska uczestnicy otrzymują po zapisie.

Metody nauczania

Warsztaty (~65%) oraz część wykładowa (~35%). Każdy blok praktyczny kończy się krótkim retro i wymianą doświadczeń. Uczestnicy wychodzą z gotowym kodem i historią promptów.

Materiały edukacyjne

- Prezentacja (PDF)
- Repozytorium Git z branchami do każdego modułu
- Prompt-notebook (Markdown)
- Board z dodatkowymi linkami i teorią

Korzyści

- Nowa kompetencja: sterowanie przeglądarką przez MCP — umiejętność coraz bardziej poszukiwana.
- Szybsze tworzenie automatyzacji dzięki wsparciu AI.
- Lepsze testy eksploracyjne: AI jako asystent.
- Świadome zarządzanie ryzykiem: checklista i wskazówki, kiedy nie ufać LLM.
- Wymiana wiedzy: prompty, szablony i wzorce repozytoriów.