

Spis treści

AI i testowanie

- AI w testowaniu: od ChatGPT do generacji kodu testów (Poziom: podstawy) 2
- Playwright + LLM + MCP w testach API i UI (Poziom: zaawansowany) 4

Testowanie UI

- Playwright - podstawy (Poziom: podstawy) 6
- Playwright - średniozaawansowany: stabilność, skalowanie, CI (Poziom: średni 7
- Cypress + JavaScript - podstawy (Poziom: podstawy) 8
- Cypress zaawansowany: izolacja, TypeScript, CI (Poziom: zaawansowany) 9
- Selenium WebDriver dla średniozaawansowanych: budowa frameworku (Pozi 10

Testowanie wydajności

- Testy wydajnościowe z Gatling + Java (Poziom: średniozaawansowany) 12
- Testy wydajnościowe w k6 + JavaScript (Poziom: średniozaawansowany) 13

CI/CD

- CI/CD i obserwowalność testów (Poziom: średniozaawansowany) 14

Strona: awesome-testing.com

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

AI w testowaniu: od ChatGPT do generacji kodu testów

Praktyczne szkolenie o prompt engineeringu, AI-assisted test design i codziennych workflow QA z ChatGPT oraz nowoczesnymi IDE z AI.

Poziom: podstawy

Czas trwania: 3 dni

Tryb: zdalnie, stacjonarnie

Języki: PL, EN

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

Szczegóły: <https://awesome-testing.com/szkolenia/ai-chatgpt-test-code>

Najważniejsze efekty

- Projektowanie skutecznych promptów do zadań testowych i analizy błędów
- Generowanie i walidacja scenariuszy testowych oraz danych testowych z AI
- Bezpieczne wykorzystanie asystentów AI w kodowaniu i review

Pełny program

Cel szkolenia

AI staje się kluczowym elementem nowoczesnego wytwarzania oprogramowania. To szkolenie wyposaża testerów w praktyczną wiedzę i umiejętności efektywnego korzystania z narzędzi takich jak ChatGPT oraz IDE wspierane AI (np. Cursor) w codziennej pracy QA.

Po ukończeniu szkolenia uczestnik będzie potrafił:

- zrozumieć, jak działa model językowy (LLM) taki jak ChatGPT,
- tworzyć skuteczne prompty przyspieszające pracę testową,
- świadomie wykorzystywać AI przy tworzeniu i przeglądzie kodu testów automatycznych.

Zakres tematyczny

- Podstawy działania LLM i praktyczne rozumienie ChatGPT
- Kluczowe pojęcia: AI, NLP, LLM, prompt
- Mechanika modeli: tokenizacja, embeddingi, attention mechanism
- Architektura Transformer i podstawy trenowania modeli
- Prompt engineering: techniki i dobre praktyki
 - role prompting (act as)
 - few-shot prompting
 - chain-of-thought
 - challenge/critique prompting
 - dobór języka promptowania
- Zastosowania AI w testowaniu
 - generowanie danych testowych
 - generowanie przypadków testowych
 - pair-testing z AI

- wykorzystanie AI do szybszego uczenia się i analizy
- Ćwiczenie praktyczne: generowanie konfiguracji CI (GitHub Actions)
- Workflow z IDE wspieranym AI (Cursor i podobne)
 - wsparcie AI przy pisaniu i refaktoryzacji testów
 - wyszukiwanie semantyczne i praca z dokumentacją
- Warsztaty end-to-end
 - tworzenie testów automatycznych z Playwright i JavaScript/TypeScript
 - ocena: co delegować do AI, a co robić manualnie
- Ryzyka i kontrola jakości
 - typowe błędy i halucynacje modeli
 - bezpieczne wzorce pracy z AI w QA
- Kolejne kroki rozwoju, w tym praca z API modeli AI

Jak się przygotować

Dla kogo?

Szkolenie jest przeznaczone dla testerów i inżynierów QA, którzy znają podstawy programowania (preferowany JavaScript).

Co przygotować?

Uczestnicy powinni mieć własny laptop przygotowany zgodnie z instrukcją trenera przekazaną przed szkoleniem.

Metody nauczania

Szkolenie ma przede wszystkim charakter warsztatowy, uzupełniony krótkimi blokami teoretycznymi. Uczestnicy uczą się przez praktyczne ćwiczenia i wspólną implementację.

Materiały edukacyjne

- prezentacja szkoleniowa,
- gotowe przykłady kodu (podzielone na moduły/branche),
- notatki robocze i linki referencyjne,
- dodatkowe materiały pomocnicze używane podczas warsztatów.

Korzyści

- Praktyczne zrozumienie działania LLM w kontekście testowania.
- Umiejętność tworzenia promptów poprawiających jakość i użyteczność wyników.
- Lepsze wykorzystanie IDE wspieranych AI przy tworzeniu i utrzymaniu testów.
- Doświadczenie praktyczne w generowaniu i walidacji testów z pomocą AI.
- Świadomość ryzyk oraz wdrożenie bezpiecznych praktyk pracy zespołowej.
- Powtarzalny workflow, który można od razu zastosować w realnych projektach QA.

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.

Strona: awesome-testing.com

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

Playwright - podstawy

Od zera do produktywnego tworzenia testów Playwright w TypeScript, z debuggingiem i fundamentami pod CI.

Poziom: podstawy

Czas trwania: 2 dni

Tryb: zdalnie, stacjonarnie

Języki: PL, EN

Narzędzia: Playwright, JavaScript, TypeScript

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

Najważniejsze efekty

- Budowa przejrzystej struktury projektu Playwright
- Pisanie stabilnych testów UI i podstawowych API
- Debugowanie awarii na podstawie trace i artefaktów

Pełny program

Opis

Praktyczne szkolenie wprowadzające, które prowadzi od pierwszych testów w Playwright do przejrzystej i utrzymywalnej bazy automatyzacji gotowej do pracy zespołowej i uruchomień CI.

Czego się nauczysz

- Zbudować projekt Playwright od zera z czytelną strukturą
- Tworzyć stabilne testy UI i podstawowe testy API
- Wykorzystywać trace, screenshoty i logi do debugowania
- Przygotować suite pod podstawowe uruchomienia CI

Agenda szkolenia

Dzień 1: Core workflow Playwright

- Instalacja i start projektu
- Budowa testu, selektory i asercje
- Contexty przeglądarki, waits i podstawy stabilności
- Pierwsze scenariusze end-to-end

Dzień 2: Utrzymanie i gotowość zespołowa

- Reużywalne helpery, fixtures i organizacja kodu
- Praca z danymi i konfiguracją środowisk
- Debugowanie awarii na podstawie artefaktów
- Konfiguracja pod CI i praktyczne konwencje zespołowe

Wymagania wstępne

Rekomendowana jest podstawowa wiedza testowa. Doświadczenie z Playwright nie jest wymagane.

Strona: awesome-testing.com

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

Playwright - średniozaawansowany: stabilność, skalowanie, CI

Optymalizacja flaky suite, lepsza architektura z fixtures oraz skalowanie wykonania przez parallelism i sharding.

Poziom: średniozaawansowany

Czas trwania: 2 dni

Tryb: zdalnie, stacjonarnie

Języki: PL, EN

Narzędzia: Playwright, JavaScript, TypeScript, GitHub Actions

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

Najważniejsze efekty

- Redukcja flaków dzięki trace-driven diagnosis
- Refaktoryzacja architektury testów wokół fixtures
- Skalowanie czasu wykonania bez utraty stabilności w CI

Pełny program

Opis

Szkolenie dla zespołów, które mają już automatyzację w Playwright i chcą zwiększyć stabilność, poprawić architekturę testów oraz skrócić czas wykonania w CI. Program koncentruje się na redukcji flake i skalowaniu dostarczania testów.

Czego się nauczysz

- Diagnozować i redukować flaky testy na podstawie trace i artefaktów
- Refaktoryzować architekturę testów z użyciem fixtures i reużywalnych wzorców
- Skalować wykonanie przez parallelism, sharding i strategię środowisk
- Poprawiać jakość feedbacku z CI i utrzymywalność suite

Agenda szkolenia

Dzień 1: Stabilność i architektura

- Przyczyny niestabilności w dojrzałych suite testowych
- Debugowanie trace-driven i klasyfikacja awarii
- Wzorce fixtures dla czytelnego i utrzymywalnego kodu testów
- Zarządzanie danymi testowymi i stanem aplikacji na większą skalę

Dzień 2: Skalowanie w CI

- Optymalizacja czasu wykonania i strategia parallel execution
- Sharding, retry i selektywne uruchamianie testów
- Diagnostyka CI i szybkie pętle feedbacku
- Governance dla długoterminowego zdrowia suite

Wymagania wstępne

Wymagane jest praktyczne doświadczenie z Playwright oraz podstawowa znajomość CI.

Strona: awesome-testing.com

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

Cypress + JavaScript - podstawy

Praktyczne szkolenie Cypress skupione na realnym setupie projektu, testach UI/API i efektywnym workflow JavaScript.

Poziom: podstawy

Czas trwania: 2 dni

Tryb: zdalnie, stacjonarnie

Języki: PL, EN

Narzędzia: Cypress, JavaScript, TypeScript

Szczegóły: <https://awesome-testing.com/szkolenia/cypress-javascript-foundations>

Najważniejsze efekty

- Setup kompletnego środowiska testowego Cypress
- Pisanie utrzymywanych testów end-to-end i API
- Praca z nowoczesnym toolchainem front-end testów na co dzień

Pełny program

Opis

Praktyczne szkolenie, które prowadzi od konfiguracji projektu do tworzenia utrzymywanych testów w Cypress. Uczestnicy pracują na realistycznych scenariuszach UI i API oraz budują efektywny workflow automatyzacji na co dzień.

Czego się nauczysz

- Konfigurować projekt Cypress z przejrzystą strukturą testów w JavaScript
- Pisać utrzymywane testy UI oraz API
- Skutecznie pracować z selektorami, fixture i danymi testowymi
- Uruchamiać testy lokalnie i w podstawowym workflow CI

Agenda szkolenia

Moduł 1: Fundamenty i konfiguracja

- Architektura Cypress i model wykonania testów
- Konfiguracja środowiska i start projektu
- Struktura testów, nazewnictwo i podstawy utrzymania

Moduł 2: Praktyka UI i API

- Stabilne interakcje UI i asercje
- Podstawy testowania API i walidacja odpowiedzi
- Praca z fixture, mockami i helperami wielokrotnego użycia

Moduł 3: Workflow i jakość

- Debugowanie i analiza awarii testów
- Prosta integracja z CI i raportowanie
- Najczęstsze antywzorce i sposoby stabilizacji

Wymagania wstępne

Rekomendowana jest podstawowa znajomość JavaScript. Doświadczenie z Cypress nie jest wymagane.

Strona: awesome-testing.com

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

Cypress zaawansowany: izolacja, TypeScript, CI

Zaawansowane szkolenie Cypress obejmujące testy izolowane, TypeScript, CI, zarządzanie flake oraz opcjonalny visual testing.

Poziom: zaawansowany

Czas trwania: 2 dni

Tryb: zdalnie, stacjonarnie

Języki: PL, EN

Narzędzia: Cypress, JavaScript, TypeScript, GitHub Actions, Percy

Szczegóły: <https://awesome-testing.com/szkolenia/cypress-advanced>

Najważniejsze efekty

- Wdrożenie solidnych praktyk testowania UI w izolacji
- Integracja uruchomień Cypress z workflow CI/CD
- Użycie zaawansowanych wzorców skalowania i niezawodności

Pełny program

Opis

Zaawansowane szkolenie dla zespołów, które już pracują w Cypress i chcą zwiększyć stabilność, utrzymywalność oraz skalowalność automatyzacji UI. Warsztat skupia się na eliminacji flake, jakości architektury testów i efektywnym uruchamianiu w CI.

Czego się nauczysz

- Tworzyć izolowane i deterministyczne testy w Cypress
- Stosować TypeScript dla lepszej jakości kodu testowego
- Integrować suite Cypress z wydajnym workflow CI
- Ograniczać flaky testy dzięki skutecznej diagnostyce

Agenda szkolenia

Dzień 1: Architektura i niezawodność testów

- Zasady izolacji i kontrola stanu testów
- Spójna strategia testów komponentowych i end-to-end
- Struktura projektu TypeScript pod reużywalność
- Wzorce dla selektorów, fixture i danych testowych

Dzień 2: Skalowanie CI i jakość operacyjna

- Optymalizacja pipeline'u pod czas i stabilność
- Diagnostyka flake, retry i zasady kwarantanny testów
- Parallel execution i orkiestracja środowisk
- Opcjonalna integracja visual regression i jej governance

Wymagania wstępne

Wymagane jest praktyczne doświadczenie z Cypress oraz podstawowa znajomość pracy z CI.

Strona: awesome-testing.com

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

Selenium WebDriver dla średniozaawansowanych: budowa frameworku

Praktyczne szkolenie z budowy utrzymywalnego frameworku Selenium: architektura, wzorce, raportowanie i uruchomienia z Selenium Grid.

Poziom: średniozaawansowany

Czas trwania: 2 days

Tryb: zdalnie, stacjonarnie

Języki: PL, EN

Narzędzia: Selenium, Java, Selenium Grid

Szczegóły: <https://awesome-testing.com/szkolenia/selenium-framework-intermediate>

Najważniejsze efekty

- Design and build maintainable Selenium framework modules
- Apply Page Object and test-data patterns to create reusable tests
- Run, debug, and troubleshoot tests locally and in remote environments (including Selenium Grid)

Pełny program

Cel szkolenia

Selenium WebDriver to darmowe narzędzie służące do automatyzacji aplikacji webowych.

Celem szkolenia jest rozszerzenie wiedzy z zakresu Selenium WebDriver oraz praktyczne zbudowanie utrzymywalnego frameworku testowego. Uczestnicy poznają sposoby optymalizacji pracy, zaawansowane techniki testowania oraz najlepsze praktyki tworzenia testów automatycznych.

Szkolenie jest realizowane głównie w Javie. Dla grup zamkniętych możliwa jest realizacja także w Pythonie, C# lub JavaScriptcie. Kurs ma charakter warsztatowy — każdy temat kończy się implementacją elementów frameworku.

Zakres tematyczny

- Selenium WebDriver: rozwinięcie podstawowych pojęć i technik
- Architektura i budowa frameworku
- Struktura projektu
- Tworzenie bazowych komponentów (Framework Core)
- Konfiguracja testów i ich uruchamianie
 - Properties
 - Parameters
- Parametryzacja testów danymi testowymi
 - DataFactory
 - DataReader
- Narzędzia pomocnicze
 - Utils
- Automatyczne logowanie przebiegu testów
- Automatyczne raportowanie testów
- Automatyczne wykonywanie zrzutów ekranu

- Integracja z zewnętrznymi bibliotekami i narzędziami
- Wzorce projektowe
 - Page Object Pattern i PageFactory (zaawansowane zastosowania)
- Uruchamianie zdalne
 - RemoteWebDriver i Selenium Grid
 - Konfiguracja serwera i węzłów
 - Uruchamianie testów na zdalnych maszynach
- Rozwiązywanie typowych problemów związanych z frameworkiem
- Testowanie aplikacji z wykorzystaniem stworzonego frameworku

Jak się przygotować

Dla kogo?

Szkolenie jest przeznaczone dla osób, które:

- programują w Javie, JavaScriptcie lub Pythonie na poziomie podstawowym, oraz
- znają podstawy Selenium WebDriver.

Udział będzie odpowiedni dla testerów manualnych rozwijających się w stronę automatyzacji, testerów automatyzujących na poziomie junior/regular oraz programistów.

Przygotowanie

Na szkolenie należy przygotować własny laptop zgodnie z instrukcją przekazaną przez trenera.

Metody nauczania

Szkolenie ma w większości charakter warsztatowy, uzupełniony o część wykładową. Uczestnicy uczą się poprzez rozwiązywanie zadań praktycznych.

Materiały edukacyjne

Uczestnicy otrzymują komplet materiałów szkoleniowych (prezentacja oraz materiały pomocnicze) oraz dostęp do materiałów elektronicznych na platformie edu.ittraining.pl.

Korzyści

- Rozszerzysz wiedzę z zakresu testów automatycznych z Selenium WebDriver.
- Poznasz zaawansowane techniki i możliwości narzędzia.
- Nauczysz się tworzyć własny framework do testów aplikacji webowych lub rozbudowywać istniejący.
- Zwiększysz konkurencyjność na rynku pracy i wzmocnisz rozwój kariery w kierunku automatyzacji testowania.

Ścieżka dalszego rozwoju

Możliwa jest rekomendacja dalszej ścieżki rozwoju kompetencji poprzez powiązane kursy z obszaru Selenium. Udział w innych szkoleniach nie jest wymagany.

Strona: awesome-testing.com

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

Testy wydajnościowe z Gatling + Java

Szkolenie skupione na metodyce, modelowaniu ruchu, implementacji w Gatling oraz raportowaniu wspierającym decyzje.

Poziom: średniozaawansowany

Czas trwania: 2 dni

Tryb: zdalnie, stacjonarnie

Języki: PL, EN

Narzędzia: Gatling, Java

Szczegóły: <https://awesome-testing.com/szkolenia/performance-gatling-java>

Najważniejsze efekty

- Planowanie testów wydajnościowych od celów biznesowych
- Implementacja solidnych scenariuszy w Gatling + Java
- Przygotowanie raportów SLA/SLO dla interesariuszy technicznych i biznesowych

Pełny program

Opis

Praktyczne szkolenie z planowania, implementacji i analizy testów wydajnościowych w Gatling i Java. Program łączy metodykę performance testów z ćwiczeniami skryptowymi i interpretacją raportów pod decyzje SLA/SLO.

Czego się nauczysz

- Przekładać cele biznesowe na mierzalne cele testów wydajnościowych
- Tworzyć reużywalne scenariusze obciążenia w Gatling + Java
- Modelować realistyczny ruch i dane dynamiczne
- Komunikować wyniki czytelnie dla odbiorców technicznych i biznesowych

Agenda szkolenia

Dzień 1: Metodyka i projektowanie scenariuszy

- Cele i zakres testów wydajnościowych oraz analiza ryzyka
- Modelowanie obciążenia i zachowań użytkowników
- Struktura projektu Gatling i fundamenty skryptów
- Asercje i progi zgodne z celami usługi

Dzień 2: Realizm i raportowanie

- Korelacja, tokeny dynamiczne i feedery danych
- Utwardzanie scenariuszy pod powtarzalne uruchomienia
- Analiza raportów: opóźnienia, throughput, błędy, nasycenie
- Przekład wyników technicznych na rekomendacje release

Wymagania wstępne

Rekomendowana jest podstawowa znajomość API/HTTP oraz podstawy Java.

Strona: awesome-testing.com

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

Testy wydajnościowe w k6 + JavaScript

Projektowanie i wykonywanie praktycznych testów obciążeniowych w k6 z realistycznymi danymi, korelacją i raportowaniem dla biznesu.

Poziom: średniozaawansowany

Czas trwania: 2 dni

Tryb: zdalnie, stacjonarnie

Języki: PL, EN

Narzędzia: k6, JavaScript, TypeScript

Szczegóły: <https://awesome-testing.com/szkolenia/performance-k6-javascript>

Najważniejsze efekty

- Budowa reużywalnych scenariuszy testów wydajnościowych w k6
- Użycie realistycznych feederów i korelacji tokenów
- Interpretacja metryk i komunikowanie wyników interesariuszom

Pełny program

Opis

Szkolenie uczy praktycznego prowadzenia testów wydajnościowych w k6 z użyciem JavaScript: od projektu scenariusza, przez wykonanie, po raportowanie pod decyzje biznesowe. Uczestnicy budują reużywalne skrypty i uczą się interpretować kluczowe metryki.

Czego się nauczysz

- Projektować skuteczne scenariusze load i stress w k6
- Tworzyć reużywalne skrypty wydajnościowe w JavaScript
- Pracować z realistycznymi danymi i korelacją tokenów
- Interpretować metryki i formułować rekomendacje

Agenda szkolenia

Dzień 1: Fundamenty i implementacja

- Podstawowe pojęcia performance i definiowanie celów
- Konfiguracja k6, struktura projektu i cykl życia skryptu
- Modelowanie ścieżek użytkownika i rozkładu ruchu
- Asercje, progi i strategia uruchomień bazowych

Dzień 2: Zaawansowane uruchomienia i analiza

- Korelacja i obsługa danych dynamicznych
- Parametryzacja środowisk i powtarzalne wykonania
- Głęboka analiza opóźnień, błędów i throughput
- Raportowanie dla interesariuszy i priorytety optymalizacji

Wymagania wstępne

Rekomendowana jest podstawowa znajomość JavaScript oraz fundamentów web/API.

Strona: awesome-testing.com

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

CI/CD i obserwowalność testów

Praktyczne szkolenie CI/CD dla pipeline testowych, uruchomień kontenerowych, quality gates i diagnostyki awarii.

Poziom: średniozaawansowany

Czas trwania: 2 dni

Tryb: zdalnie, stacjonarnie

Języki: PL, EN

Narzędzia: GitHub Actions, Docker, Jenkins

Szczegóły: <https://awesome-testing.com/szkolenia/cicd-observability>

Najważniejsze efekty

- Budowa pipeline testowych w GitHub Actions i Jenkins
- Konteneryzacja uruchomień testów dla powtarzalności
- Wdrożenie obserwowalności i praktycznych quality gates

Pełny program

Opis

Praktyczne szkolenie o budowie niezawodnych pipeline'ów testowych z wykorzystaniem CI/CD, konteneryzacji i obserwowalności. Program koncentruje się na powtarzalnym uruchamianiu testów, szybkiej diagnostyce i quality gates wspierających decyzje release.

Czego się nauczysz

- Budować i porządkować pipeline'y testowe w GitHub Actions i Jenkins
- Konteneryzować uruchomienia testów dla pełnej powtarzalności
- Wdrażać praktyczne quality gates i proces triage awarii
- Zwiększać widoczność stanu pipeline'u i jakości suite testowych

Agenda szkolenia

Dzień 1: Architektura pipeline CI/CD

- Projektowanie etapów pipeline pod testy i środowiska
- Wzorce uruchomień testów w Dockerze
- Strategia triggerów dla PR, nightly i release
- Artefakty, retry oraz stabilizacja podstawowego flow

Dzień 2: Obserwowalność i kontrola jakości

- Zbieranie logów, trace, screenshotów i raportów
- Analiza trendów wyników testów
- Quality gates oparte o ryzyko, pokrycie i pewność wydania
- Triage awarii buildów i flaky testów w stylu incident response

Wymagania wstępne

Wskazana jest podstawowa znajomość automatyzacji testów i pracy z repozytorium kodu.