

Program szkolenia:

Testy wydajnościowe w JMeter w środowisku Cloud z CI/CD

Informacje:

Nazwa:	Testy wydajnościowe w JMeter w środowisku Cloud z CI/CD
Kod:	cloud-jm
Kategoria:	Narzędzia
Odbiorcy:	
Czas trwania:	3 dni
Forma:	20% wykłady / 80% praktyka

Celem szkolenia jest zdobycie przez uczestnika umiejętności praktycznych z dziedziny testów wydajnościowych z użyciem popularnego narzędzia Apache JMeter.

Podczas szkolenia przedstawione są zagadnienia związane z teorią testów, kierunkiem ich rozwoju oraz przyszłością, które pozwalają uczestnikom zrozumieć szerszą pojętą dyscyplinę Performance Engineering' u. Kurs jest nastawiony na praktykę (80%) z niezbędnym udziałem części teoretycznej (20%).

Podczas szkolenia wykonujemy testy na różnego rodzaju aplikacjach, z którymi spotykamy się w codziennej pracy - tradycyjnych oraz typu SPA (Single-Page Application). Poznajmy również dobre praktyki i nowoczesne metody integracji testów wydajnościowych z CI/CD w środowisku chmurowym.

Po ukończonym szkoleniu uczestnik powinien być w stanie samodzielnie pisać testy wydajnościowe dla aplikacji webowych.

Zalety szkolenia:

- Kurs jest nastawiony na praktykę z niezbędnymi elementami teoretycznymi
- Poruszamy nie tylko podstawy samego narzędzia JMeter ale również jego zastosowanie do testów w środowiskach chmurowych (na przykładzie Azure)
- Omawiamy bieżące trendy w obszarze testów wydajnościowych i źródła wiedzy niezbędne w dalszym rozwoju
- Wykonujemy ćwiczenia praktyczne z zakresu integracji testów wydajnościowych z CI/CD
- Implementujemy testy z wykorzystaniem Kubernetesa/Dockera

Szczegółowy program:

1. Teoria testów wydajnościowych, ich rodzaje i zastosowanie

1.1. Rodzaje testów i nomenklatura: load, soak, stress, peak testy

1.2. Rodzaje AUT (Application-Under-Test): SPA, MPA, Low-Code, No-Code.

1.3. Analiza wymagań niefunkcjonalnych, dobór właściwej strategii testowania, najczęściej spotykane przypadki

2. Trendy w dziedzinie Performance Engineering'u jak się do tego ma Performance Testing

2.1. Shift left (k6, locust, DevWeb)

2.2. AIOps

2.3. Observability

2.4. Testy wydajnościowe w CI/CD

2.5. Testy wydajnościowe z użyciem przeglądarki: WDS i Lighthouse

3. Bliższe spojrzenie na AUT. Zapoznanie się z aplikacjami, które będziemy testować:

3.1. Strona statyczna i strona z AJAX/HXR.

3.2. Aplikacja SPA (Single Page Application)

3.3. Wybrana webowa strona publiczna

3.4. API z uwierzytelnianiem

4. Podstawy JMetera i jego architektura

4.1. Z czego składa się JMeter

4.2. Jak rozszerzyć jego funkcjonalność z użyciem pluginów

4.3. Testy rozproszone

5. Piszemy pierwsze żądania HTTP w JMeterze

5.1. Niezbędne elementy planu w JMeter

5.2. Użycie HTTP Request Samplera i ewaluacja rezultatów

5.3. Listenery - View Results Tree

6. Różne rodzaje elementów konfiguracyjnych i zmienne użytkownika

- 6.1. Rodzaje elementów w JMeter i kolejność ich wywołania
- 6.2. Użycie User Defined Variables, HTTP Cookie Manager, HTTP Cache Manager
- 6.3. Zastosowanie elementów wprowadzających wartości domyślne - HTTP Request Defaults
- 6.4. Różnica między zmiennymi, zmiennymi użytkownika i parametrami

7. Kontrolery logiki w JMeterze, Timery

- 7.1. Kontrolery IF i LOOP
- 7.2. Pisanie warunków logicznych w JS, Groovy i Beanshellu
- 7.3. Rodzaje Timerów
- 7.4. Test Fragments, Simple Controller, Module Controller - DRY (Do-not-Repeat-Yourself) Trio

8. Procesory i Samplery JSR223, elementy skryptowalne w JMeter

- 8.1. Funkcje w JMeterze na wybranych przykładach

9. Podstawy API JMeter'owego i kilka słów o Groovy

- 9.1. Jak poruszać się po API JMeterowym
- 9.2. Wprowadzenie do języka Groovy

10. Struktura planu testowego

- 10.1. SetUpThreadGroup, TearDownThreadGroup

11. Urealistycznianie naszego scenariusza:

- 11.1. Symulacja zachowania przeglądarki
- 11.2. Dwa podejścia do modelowania zachowania użytkownika

12. Ekstraktory

- 12.1. Omówienie rodzajów ekstraktorów i ich zastosowanie
- 12.2. Boundary Extractor, XpathExtractor, CSSExtractor, Regular Expression Extractor
- 12.3. Testowanie ekstraktorów w Google Chrome i JMeter

13. Nagrywanie ruchu HTTP i jego odtwarzanie

14. Uruchamianie testów i generowanie raportów

14.1. Jak pracować podczas developmentu, gdzie trzymać kod?

14.2. Uruchamianie testów z linii poleceń i z użyciem dockera

14.3. Generowanie raportów i "lepszy" sposób na wizualizację danych - backend listenery

15. Debugowanie testów

15.1. Debugowanie skryptów w JMeter (LogLevel, logi)

15.2. Debugowanie kodu JMetera

16. Dobre praktyki na początek

16.1. Zalecane praktyki przez twórców JMetera

16.2. Utrzymywalność testów

17. Wprowadzenie do Ciągłego Testowania wydajności i integracji z CI

17.1. Uruchamianie testów JMeterowych na Jenkinsie z wykorzystaniem Mavena lub na Azure Cloud

17.2. Nowoczesne podejście do testowania wydajności w chmurze z Dockerem i Kubernetes

17.3. Narzędzia, które warto znać

18. Hackathon lub Tematy zaawansowane

19. Skąd czerpać wiedzę