

Testy wydajnościowe aplikacji webowych i usług z wykorzystaniem Apache JMeter

Tworzenie skryptów i przeprowadzanie testów

Cele szkolenia

Celem szkolenia jest wprowadzenie uczestnika w tematykę testów wydajnościowych i ich wykonywania z wykorzystaniem narzędzia JMeter. Podczas szkolenia uczestnicy poznają narzędzie JMeter oraz nauczą się samodzielnie przygotowywać skrypty testowe, zarówno do testów aplikacji webowych jak i API. Podczas szkolenia uczestnicy będą wspólnie z prowadzącą przygotowywali skrypt do testów rzeczywistej aplikacji webowej i usługi.

Umiejętności

Dzięki szkoleniu uczestnik będzie:

- Przygotowywał skrypty do testów wydajnościowych dla aplikacji webowej i usługi (REST, SOAP).
- Dostosowywał parametry skryptu wydajnościowego do specyfiki systemu.
- Wykonywał testy wydajnościowe i potrafił zinterpretować ich wyniki oraz wskazać problemy wydajnościowe systemu.
- Przygotowywał czytelny raport z wynikami testów.
- Wykonywał testy w trybie GUI oraz z linii poleceń.
- Wykonywał testy z wielu maszyn w trybie testowania rozproszonego (Distributed Testing).
- Potrafił przechwytywać żądania HTTP z wykorzystaniem narzędzia typu lokalnego proxy.
- Parametryzował skrypty z wykorzystaniem wyrażeń regularnych.
- Wykorzystywał popularne rozszerzenia do narzędzia JMeter.

Profil uczestników

Szkolenie jest skierowane dla testerów zaczynających przygodę z testami wydajności z wykorzystaniem narzędzia JMeter. Uczestniczenie w szkoleniu wymaga podstawowej wiedzy z obszaru testów aplikacji webowych.

Przygotowanie uczestników

Instalacja Javy i narzędzia JMeter oraz aplikacji testowych. Aplikacje testowe wymagają instalacji Dockera.

Szczegółowy program szkolenia

1. Wprowadzenie
2. Podstawy testów wydajnościowych
3. Rodzaje testów wydajnościowych m. in.
 - 3.1. Baseline Test
 - 3.2. Load Test
 - 3.3. Soak Test
 - 3.4. Stress Test
 - 3.5. Volume Test
4. Strategie testowania wydajności
5. Podział testów ze względu na cel testu
6. Analiza wymagań pod kątem wydajności
7. Podstawy komunikacji HTTP
8. Wprowadzenie do JMetera i podstawy korzystania z narzędzia
9. Podstawowe elementy tworzące plan testów m. in:
 - 9.1. Test Plan
 - 9.2. Thread Group
 - 9.3. Sampler
 - 9.4. Listener
 - 9.5. Timer
 - 9.6. Assertion
10. Przygotowanie środowiska do testów wydajnościowych (zasilenie środowiska przed i w trakcie testów, import lub pobranie danych testowych)
 - 10.1. JDBC Sampler
11. Modelowanie zachowania użytkowników poprzez kontrolery i niestandardowe grupy wątków
 - 11.1. Plugin Ultimate Thread Group i zmienna liczba wątków/wirtualnych użytkowników w trakcie testu
 - 11.2. Modelowanie zachowań użytkowników
 - 11.3. Kontrolery - instrukcje warunkowe na przykładzie:
 - 11.3.1. While Controller
 - 11.3.2. If Controller
 - 11.3.3. Once Only Controller
 - 11.3.4. For Each Controller
 - 11.4. Inne ważne kontrolery
 - 11.4.1. Module Controller i Test Fragment - wielokrotne wykorzystanie fragmentów skryptu
 - 11.4.2. Throughput Controller i Weighted Switch Controller - zarządzanie ruchem
12. Parametryzacja na przykładzie zmiennych dynamicznych np. tokeny anty-CSRF - wykorzystanie RegEx (Regular Expression) Extractor
13. Skrypty (Groovy/BeanShell/JavaScript/Python)
 - 13.1. Jak pisać skrypty nie tylko w groovy, czyli konfiguracja skryptów python
14. Uruchamianie testów z wiersza poleceń i generowanie raportu HTML
15. APDEX (Application Performance Index) i jego konfiguracja w JMeterze
16. Dodatkowe pluginy - Plugin Manager
 - 16.1. Testy usług REST
 - 16.2. REST i JWT
 - 16.3. REST i OAuth
 - 16.4. Authorization Manager
17. Testy usług Soap
 - 17.1. Przygotowanie projektu testowego na podstawie pliku WSDL
 - 17.2. Parametryzacja i przygotowywanie unikalnych danych testowych z wykorzystaniem skryptów groovy
 - 17.3. Konfiguracja WS-Security
 - 17.4. "Przeniesienie" skryptów SoapUI do JMetera
18. Upload plików - konfiguracja i problemy które mogą się pojawić

19. Monitorowanie zużycia zasobów na serwerze podczas trwania testów wydajnościowych na przykładzie pluginu SSHMon
20. Testy z wykorzystaniem wielu maszyn (Distributed Testing) - konfiguracja środowiska i problemy które mogą się pojawić
21. Debugowanie skryptów (Debug sampler, narzędzia proxy, Dummy Sampler)
22. JMeter to nie wszystko - kilka słów o wydajności frontendu i jak ją poprawić (HTTP 2, WebPageTest)

Metoda realizacji szkolenia

Zdalnie lub stacjonarnie. Szkolenie prowadzone w formie warsztatowej z ćwiczeniami praktycznymi.

Liczba dni, liczba godzin szkoleniowych

2 dni, 14 godzin szkoleniowych

Ścieżka rozwoju po szkoleniu

Zachęcamy do zapoznania się z ofertą szkoleń dotyczących testowania.