

## Program szkolenia:

# Testy wydajności z Gatlingiem

### Informacje:

<b>Nazwa:</b>	<b>Testy wydajności z Gatlingiem</b>
<b>Kod:</b>	<b>craft-test-gat</b>
<b>Kategoria:</b>	Testowanie automatyczne
<b>Odbiorcy:</b>	testerzy, developerzy
<b>Czas trwania:</b>	2 dni
<b>Forma:</b>	25% wykłady / 75% warsztaty

---

Podczas szkolenia uczestnicy poznają praktyczne techniki testowania wydajności systemów z wykorzystaniem Gatlinga.

Warsztaty skupiają się głównie na praktycznym podejściu do testowania wydajności - pracujemy z realną aplikacją webową, którą należy przetestować pod kątem wydajności i przeanalizować gdzie są problemy z jej wydajnością. Między praktycznymi zadaniami poruszamy teoretyczne podstawy testowania wydajności oraz pokazujemy najczęstsze błędy oraz jak ich unikać.

Gatling jest jednym z najbardziej popularnych narzędzi do testowania wydajności. Umożliwia tworzenie bardzo czytelnych, reużywalnych i komponowalnych testów wydajności, a niektóre z funkcji nie są możliwe do osiągnięcia w innych narzędziach.

W przeciwieństwie do rozwiązań opartych na UI (JMeter), do modyfikacji i rozwijania testów można użyć wszystkich ulubionych technik refaktoryzacji i rozwijania oprogramowania.

Wymaga dużo mniej zasobów do wygenerowania oczekiwane obciążenia, a dodatkowo umożliwia rozproszone testowanie z wielu maszyn jednocześnie, co jest szczególnie istotne w przypadku systemów opartych na rozwiązaniach chmurowych. Szkolenie możliwe jest do przeprowadzenia w Javie lub Scali.

### Zalety szkolenia:

- Czytelne i utrzymywalne testy wydajności jako kod.
- Ćwiczenia praktyczne i dużo kodowania.

## Szczegółowy program:

### 1. Podstawy testowania wydajności

1.1. przygotowanie środowiska testowego

1.2. obserwowalność środowiska

1.2.1. metryki

1.2.2. logowanie

1.2.3. profilowanie

### 2. Metryki w testach wydajności

2.1. jak nie dać oszukać się wartościom statystycznym

2.2. porównywanie testów wydajności

### 3. Błędy w narzędziach do testowania wydajności

3.1. sposoby liczenia percentyli

3.2. Coordinated Omission problem

### 4. Podstawy Gatlinga

4.1. architektura

4.2. porównanie z innymi narzędziami do testowania wydajności

4.3. utworzenie i uruchomienie pierwszego scenariusza testowego

### 5. Zaawansowane funkcje Gatlinga

5.1. walidacje odpowiedzi

5.2. pętle

5.3. wyrażenie warunkowe

5.4. losowość danych

5.5. wirtualna sesja użytkownika

5.6. obsługa błędów w odpowiedziach

**6. Jakość testów**

6.1. poprawa modularności, reużywalności, kompozycji testów

6.2. zapewnienie utrzymywalności testów i gotowość na zmiany

**7. Symulacja ruchu produkcyjnego**

7.1. zaawansowane sterowanie ruchem generowanym przez Gatlinga

7.2. generowanie danych testowych

**8. Wyniki testów wydajności**

8.1. throughput a latency

8.2. generowanie raportów HTML

8.3. globalne i jednostkowe metryki oraz wykresy

**9. Testy rozproszone**

9.1. generowanie maksymalnego obciążenia z wielu maszyn jednocześnie

9.2. zbierania i przetwarzanie wyników

**10. Testy asynchroniczne**

10.1. problemy z testami asynchronicznymi

10.2. testy WebSocketów lub JMS