

Performance-Testautomatisierung mit JMeter, Gatling und Locust

Lasttests automatisieren und geeignete Werkzeuge gezielt einsetzen

01 INTRO

Performance-Tests müssen reproduzierbar sein, um Veränderungen zwischen Versionen, Konfigurationen und Lastprofilen erkennen zu können. Automatisierte Lasttests verbinden definierte Szenarien mit messbaren Schwellenwerten und ermöglichen wiederholbare Performance-Prüfungen bis hin zur Integration in Delivery Pipelines.

Der Kurs behandelt die technische Automatisierung von Last- und Performance-Tests und vergleicht dafür JMeter, Gatling und Locust. Statt drei Werkzeuge lediglich nebeneinander vorzuführen, werden ihre unterschiedlichen Modelle, Stärken und Einsatzgebiete anhand gemeinsamer Testszenarien untersucht.

DAUER

3 Tage

FORMAT

Präsenz · Live Online · Inhouse

VORKENNTNISSE

Grundlagen von Last- und Performance-Tests sowie technisches Verständnis von HTTP und APIs

02 TRAININGSPROGRAMM

Inhalte

Performance-Testautomatisierung

- Lastmodelle
- Szenarien
- virtuelle Nutzer
- Durchsatz
- Antwortzeiten
- Schwellenwerte
- reproduzierbare Tests
- Automatisierungsziele

Werkzeugauswahl

- JMeter
- Gatling
- Locust
- unterschiedliche Programmiermodelle
- GUI versus Code
- Ökosystem
- CI/CD
- Auswahlkriterien

JMeter

- Test Plan
- Thread Groups
- Sampler
- Assertions
- Timer
- Listener
- Variablen
- nichtinteraktive Ausführung

JMeter-Szenarien entwickeln

- HTTP Requests
- Parameterisierung
- CSV-Daten
- Korrelation
- Authentifizierung
- Sessions
- wiederverwendbare Komponenten

Gatling

- Simulationen
- Szenarien
- virtuelle Nutzer
- Feeder
- Checks
- Injection Profiles
- Assertions
- Reports

Locust

- Python-basierte Lasttests
- User Classes
- Tasks
- Wait Time
- Events
- Daten
- verteilte Ausführung
- Erweiterbarkeit

Gemeinsames Testszenario vergleichen

- identisches Zielsystem
- vergleichbares Lastmodell
- Implementierungsaufwand
- Lesbarkeit
- Wartbarkeit
- Ausführung
- Reporting
- Werkzeugentscheidung

Realistische Lastmodelle automatisieren

- Ramp-up
- konstante Last
- Spitzen
- unterschiedliche Benutzergruppen
- Transaktionsmix
- Think Time
- Daten
- Testdauer

Parameterisierung und Korrelation

- dynamische IDs
- Tokens
- Sessions
- abhängige Requests
- Datenquellen
- individuelle Benutzer
- reproduzierbare Abläufe

Verteilte Last und Skalierung

- mehrere Load Generators
- Controller und Worker
- Netzwerk
- Ressourcen
- Lastgenerator als möglicher Engpass
- Cloud-Ausführung einordnen
- Kapazitätsplanung

Automatisierung in CI/CD

- Kommandozeilenausführung
- Schwellenwerte
- automatischer PASS und FAIL
- Reports
- Artefakte
- kleine Regressionstests
- größere geplante Lasttests
- Trends

Praktisches Performance-Automatisierungsprojekt

- Lastmodell definieren
- Szenario implementieren
- Daten parametrisieren
- Last erzeugen
- Ergebnisse vergleichen
- Schwellenwerte definieren
- CI-Ausführung vorbereiten

03 DURCHFÜHRUNG

Rahmen und Organisation

Dauer	3 Tage
Durchführung	Präsenz oder Live Online, vorzugsweise als Inhouse-Schulung
Schwerpunkt	Inhouse-Schulungen für Unternehmen und Teams
Sprache	Deutsch, englische Fachbegriffe und Dokumentation
Unterlagen	Kursunterlagen, Beispiele und Übungsaufgaben
Technik	Eigener Rechner mit Java, Python und den eingesetzten Lastwerkzeugen

Inhouse-Schulungen

Der Kurs kann auf ein einzelnes Werkzeug konzentriert und mit dem eigenen Zielsystem durchgeführt werden.

04 ZIELGRUPPE

Für wen ist der Kurs gedacht?

Performance Tester, Testautomatisierer, Quality Engineers und Entwickler, die Lasttests reproduzierbar automatisieren möchten.

Voraussetzungen

Grundlagen von Performance Testing, HTTP und APIs. Programmiererfahrung ist hilfreich.

05 LERNZIELE

Was Sie nach dem Kurs können

Die Teilnehmer können automatisierte Lasttests entwickeln, JMeter, Gatling und Locust hinsichtlich ihrer Einsatzgebiete beurteilen und Performance-Tests reproduzierbar sowie CI-fähig gestalten.

06 FAQ

Häufige Fragen

Warum werden drei Werkzeuge behandelt?

Nicht um drei vollständige Produktschulungen in einen Kurs zu pressen, sondern um unterschiedliche Automatisierungsansätze praktisch vergleichen und eine begründete Werkzeugentscheidung treffen zu können.

Was unterscheidet den Kurs von Last- und Performance-Tests im Quality Engineering?

Der QE-Kurs behandelt **Performance Engineering und Testdesign**: Anforderungen, Lastmodelle, Messwerte und Engpassanalyse. Hier geht es um die **technische Automatisierung und die Werkzeuge** für reproduzierbare Performance-Tests.

Kann der Kurs auf ein einzelnes Werkzeug konzentriert werden?

Ja. Bei entsprechender Zielsetzung kann eines der Werkzeuge vertieft und der Vergleich entsprechend verkürzt werden.

UC IT Service · Ulrich Cuber · kontakt@uc-it.de

<https://www.uc-it.de/coaching/performance-testautomatisierung/>