

Hochschule Karlsruhe  
Technik und Wirtschaft  
UNIVERSITY OF APPLIED SCIENCES

Studiengang Informatik (Master)

# **Last-Test von Web-Seiten mit JMeter**

**Seminararbeit - Ausarbeitung**

Daniel Schäfer (60118)

**Betreuer:** Prof. Dr. Holger Vogelsang

Sommersemester 2018

# **Inhaltsverzeichnis**

# 1 Einführung

## 1.1 Motivation - Wozu überhaupt Testing?

Man erlebt es immer wieder, dass Webseiten unter zu hoher Last in die Knie gezwungen werden. Sei es nun Amazon, Ebay oder auch Hochschul interne Seiten, die durch bestimmte Anmeldefristen viele Nutzer gleichzeitig versorgen müssen. Doch warum passiert sowas in der heutigen Zeit immer noch? Und kann man dem vorbeugen und entgegenwirken? Sogenannte Last oder Performance Test sind dafür nötig. Das frei verfügbare unter Public License Key stehende Java Programm JMeter bietet diese und weitere Funktionalität um die Leistung einer Webseite bis ins kleinste Detail zu analysieren und zu testen.

## 1.2 Verschiedene Testarten

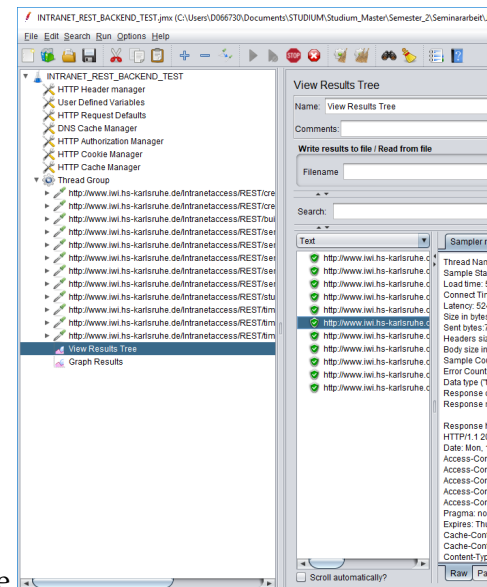
User Acceptance Testing, Compatibility Tests

## 1.3 Automatisierte und manuelle Tests

# 2 Apache JMeter

Auf den ersten Blick erscheint Apache JMeter wie das Relikt aus einer längst vergangenen Zeit. Doch man sollte sich von der Swing-basierten GUI Oberfläche nicht täuschen lassen.

Tatsächlich stammt JMeter aus einer Zeit als das Web und Application Server noch in den Kinderschuhen steckten.



Man sollte sich jedoch nicht täuschen lassen. Eierlegende

JMeter Plattformunabhängig wegen java anwendung. kann sowohl auf mac als auch auf windows laufen. lediglich die java runtime

Wann wurde das Tool das erste mal entwickelt? wie oft wird es eingesetzt. auch von großen unternehmen? kommt bei sap auch zum einsatz

## 2.1 Was kann JMeter

### 2.1.1 Lastentests

### 2.1.2 Captcha Analyse

Captcha analysieren und erkenne?

### 2.1.3 Auto vervollständigung von Formularen

befüllen von formularen

### 2.1.4 JUnit Tests

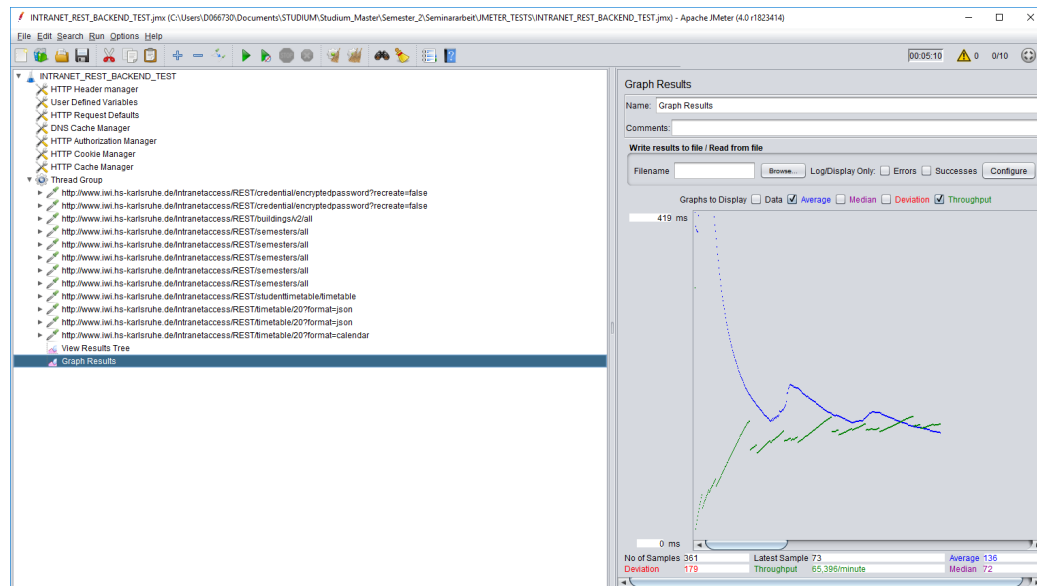
junit tests in jmeter importieren

## 2.2 Installation

### 2.2.1 Setting up the environment

### 2.2.2 Running JMeter

## 2.3 JMeter GUI



## 2.4 JMeter Command Line

GUI verbraucht viel Speicher. Nicht empfohlen für große Lastentests. Command Line ist geeignet für die Integration in andere Systeme wie Jenkins/-Continuous Integration.

### 2.4.1 Ausführen des Console Clients

Zuerst sollte ein Testscript verfügbar sein. Man kann hier ein vorhandenes File verwenden oder in JMeter selbst ein neues \*.jmx-File erzeugen. Minimale Voraussetzungen sind eine Thread Group sowie ein Sampler wie etwa ein HTTP Request mit entsprechender URI.

JMeter GUI beenden und Command Line starten. Danach in das /bin Verzeichnis von JMeter navigieren. Dort startet man den Test via dem Befehl:

```
1 jmeter -n -t [jmx file] -l [results file]
```

In der folgenden Tabelle sieht man einige häufig verwendete Parameter und ihre Bedeutung. Um eine Übersicht aller Parameter zu erhalten, kann man in der Console den `jmeter -?` ausführen.

Parametername	Bedeutung
-n	non gui mode
-t	location of jmeter script - combined with -n
-l	location of result file
-?	alle Parameter an die zur Auswahl stehen
-h	Beispielbefehle, die man verwenden kann
-L	Log level - Wann soll etwas geloggt werden
-e	Um html Reports zu erzeugen
-o	Location of the output folder - combined with -e or -g
-g	html report wird aus einem csv file erzeugt

Tabelle 1: Befehle der JMeter Command Line

## 3 Der Testplan

Was ist überhaupt ein Testplan

### 3.1 Elemente eines Testplans

Welche Elemente kann ein Testplan enthalten

1 Testplan anlegen

#### 3.1.1 Thread-Groups

Ähnlich wie JUnit. Hier Möglichkeiten `setup` `teardown` neben den eigentlichen Gruppen Gleichzeitige Benutzer usw.

#### 3.1.2 Sampler

Sind die eigentlichen Tests

#### 3.1.3 Listener

Elemente die Informationen über die Performance Tests enthalten Zum Abfragen bzw. Ergebnisse visualisieren

werden verwendet um ergebnisse und metriken der tests anzuzeigen

### 3.1.4 http request

### 3.1.5 Assertions

Mit jmeter lassen sich assertions testen. Dazu wird der response von einem request untersucht assert 200. assert 500 etc.

## 4 Lasttests von Webseiten

## 5 Funktionale Tests

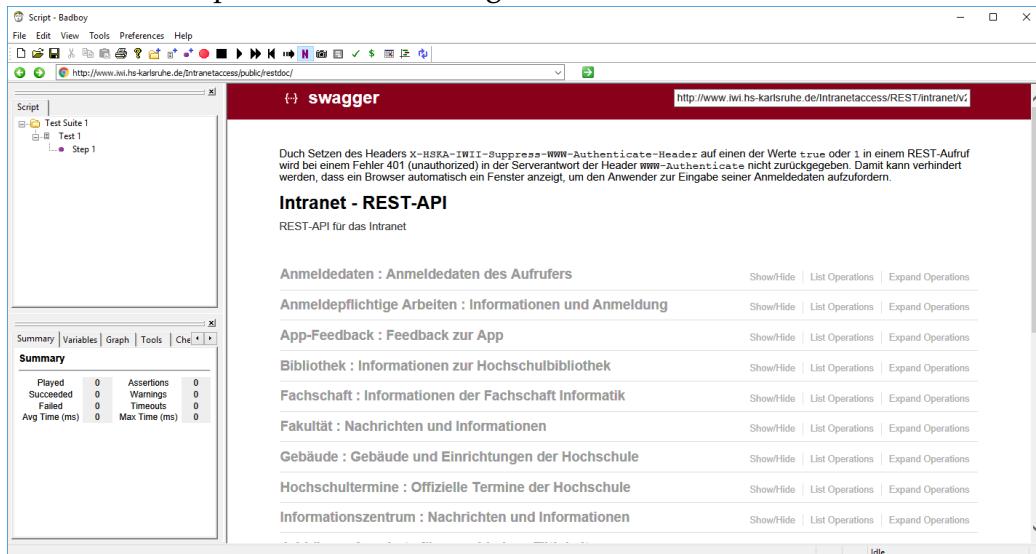
## 6 Wissenswertes und Besonderheiten

Man kann die Ergebnisse zur Laufzeit in Logfiles schreiben.

### 6.1 JMeter Plugin Manager

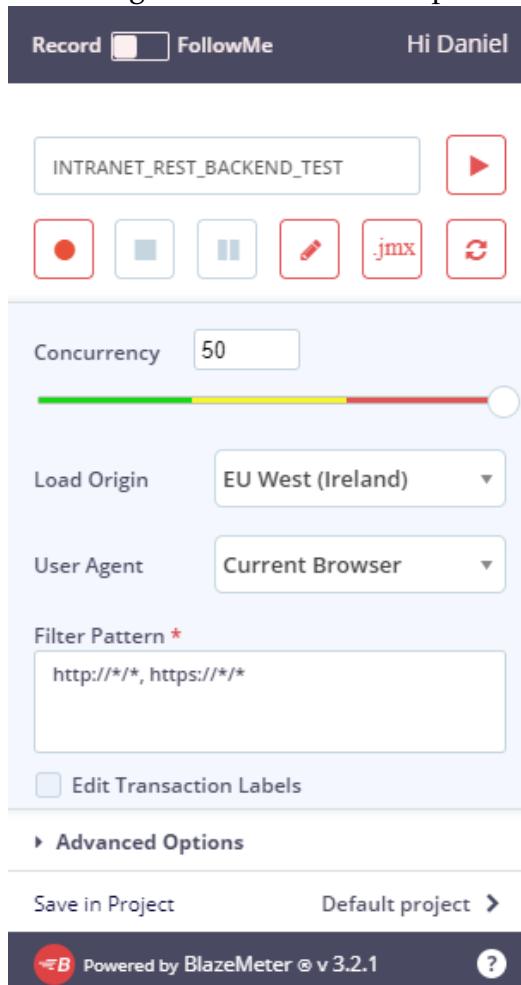
### 6.2 Aufzeichnen von Aktionen - UI web Test

Build in Solution: Bei non-test-Elements Testscript Recorder <-hiermit kann man ein script aufzeichnen um ganze Webseiten zu untersuchen



Für Windows only Plattformabhängig BadBoy: Ansonsten drittanbieter verwenden wie etwa badboy.

Chrome Plugin: Blazemeter Mit Blazemeter ist man in der Lage direkt in Chrome und somit plattformunabhängig Aufzeichnungen von Aktionen auf Webseiten aufzunehmen. Kleiner Nachteil: Man muss sich bei Blazemeter registrieren um den Export in .jmx Dateien ausführen zu können.



## 6.3 Datenbankabfragen - Datenbank testplan?

Auch datenbankabfragen lassen sich damit visualisieren

Falls interresse besteht, wie es funktioniert bitte in die Ausarbeitung schauen. Thread Gruppe erstellen -> Configuration (JDBC Connection Configuration) MySQL jdbc jar in lib folder von jmeter adden Dann Samp-



ler -> JDBC Request and put in values.. z.b. select \* from table

## **6.4 Virtuelle Compute Unit**

In die Cloud um an einer zentralen stelle viele rechner zu bedienen? Eventuell viele Rechner simulieren

## **6.5 JMeter change Settings**

In /bin folder von JMeter kann man die Datei user.properties öffnen und Werte anpassen.

## **7 Nachteile von JMeter**

Wenn ein Variablenfeld required ist. z.b. bei JDBC Connection Configuration. dann wird einem der fehler nicht direkt im feld angezeigt.

Ziemlich resourcenhungrig. Trotz 16GB Arbeitsspeicher hängt sich das Ding gerne mal auf.

## **8 Fazit**

Alles in allem ist JMeter ein Super tool