

Emma – eine kleine Webserverin in Java

Projektarbeit in Datenkommunikation von Michael Rolli

Projekt-Homepage und git-Repository: <http://github.com/mrolli/emma>

Installation/Setup

Das Eclipse-Projekt von Emma ist folgendermassen vorkonfiguriert:

- Port 8080 (http) sowie Port 8443 (https)
- Default Virtual Host: localhost (Alias: www.localhost)
- Virtual Host: test.localhost
- KeyStore für SSL wird per `System.setProperty()` in der Main-Methode gesetzt

Eintrag in `/etc/hosts`: `127.0.0.1 localhost www.localhost test.localhost`

Starten lässt sich Emma mit der Methode **ch.rollis.emma.Emma.main()**. Der Pfad zu einer alternativen Serverkonfiguration kann als erstes Argument übergeben werden, sonst wird obige Konfiguration aus `./config/server.xml` eingelesen. Details zur Konfiguration sind ebendort ersichtlich.

Eine Willkommensseite mit zusätzlichen Informationen (Features, Test Cases, API-Dokumentation) finden Sie unter <http://localhost:8080/index.html>

Für das Testing (aka Benotung) stehen Test Cases zur Verfügung. Diese sollen es ermöglichen, alle unten aufgelisteten Features erfahrbar zu machen. Diese Testseite finden Sie unter <http://localhost:8080/tests.html>.

HINWEIS: Da Emma persistierende Verbindungen ohne Pipelining unterstützt (default für HTTP/1.1), wurde entsprechend ein Request Timeout eingebaut. Dieses beträgt 15s, d.h. be quick on telnet or be disconnected! ;-) Anpassen lässt sich die Timeout-Zeit in `ch.rollis.emma.RequestHandler.requestTimeout`.

Features

- Unterstützung für **Simple-Requests** (HTTP/0.9) und **Full-Requests** (HTTP/1.0+) nach rfc1945 resp. rfc2616
- Parallele Abarbeitung aller Client-Requests (**Threading**)
- Unterstützung für persistierende Verbindungen aka **Keep-Alive** (HTTP/1.1, dort default, @see `ch.rollis.emma.RequestHandler.requestTimeout`)
- Sockets auf **mehreren Ports** möglich (default: 8080, 8443)
- Pro Port **SSL-Verschlüsselung** zuschaltbar (8443 default SSL, siehe `./config/server.xml`)
- Implementierte HTTP Methoden: **GET, HEAD** (POST, PUT, DELETE, OPTIONS nicht implementiert, aber gesendetes Entity wird geparsed und im Requestobjekt abgelegt)
- Unterstützung für **Conditional GET** mit korrektem Response Code (304 - Not Modified)

- Bei **Requests ohne trailing Slash** auf ein Directory folgt ein Redirect auf korrekte URL (301 - Moved Permanently)
- Bei Requests auf ein Directory wird in diesem nach einem konfigurierbaren (@see Emma.DEFAULT_FILES) **Index** gesucht - default: index.html, index.htm
- Bei Requests auf ein Directory ohne Index wird ein **Directory Listing** gesendet, falls dies erlaubt ist (siehe virtuelle Hosts)
- Unterstützung für **Virtuelle Hosts (VH)** (siehe `./config/server.xml`):
 - Beliebige Anzahl VH konfigurierbar
 - Pro VH ein Server Name sowie beliebige Aliase
 - **Request-Logging pro virtuellem Host** im Apache Style (inkl. Response Code, Content Length, Referer, User-Agent)
 - Auslieferung des **default Hosts**, wenn der Request einen unbekannten Host referenziert
 - Option "**allowIndexes**" (Directory Listings) pro VH

Unterstützte Header Fields: Request

- Alle Request Headers werden geparsed und im Requestobjekt abgelegt (unbekannte als Entity Headers)
- **General Headers:** Connection
- **Request Headers:** Host, If-Modified-Since (→ Conditional GET), Referer, User-Agent
- **Entity Headers:** Content-Length löst ein Einlesen des Entity aus

Unterstützte Header Fields: Response

- **General Headers:** Date (nach rfc1135), Connection
- **Response Headers:** Server, Location
- **Entity Headers:** Content-Type (Mime Type basierend auf file extensions), Content-Length, Last-Modified

Nicht zu vergessen

- **Fehlermeldungen** (z.B. 400 - Bad Request, 404 - Not found, 500 - Internal Server Error) mit **menschenlesbarer HTML-Response**
- Mittels **quit, exit oder shutdown** kann Emma per stdin gracefully beendet werden, d.h. aktuelle Requests werden noch bearbeitet, ehe alle Threads ordentlich beendet werden
- **XML-basierte Konfiguration** einiger Einstellungen des Servers und der virtuellen Hosts (**JAXB**; siehe `./config/schema.xsd`)
- Vollständige API-Dokumentation (siehe <http://localhost:8080/doc/>)

Quellenangaben

Folgende Quellen wurden bei der Erarbeitung von Emma konsultiert und für Ideen resp. Lösungsansätze herangezogen:

- **Rolf Lanz, Dozent, BFH:** Hilfestellung/Fragestunde
- **HTTP really made easy** (<http://www.jmarshall.com/easy/http/>)
- **Dokumentation Apache-2.2** (<http://httpd.apache.org/docs/2.2/>)
- **NanoHTTPD** (<http://elonen.iki.fi/code/nanohttpd/>)
- **Jibble Web Server** (<http://www.jibble.org/jibblewebserver.php>)
- **Skript „XML Technologien, SD06“, B. Amrhein, BFH:** JAXB