

U 7.1 Java Sockets: HTTP-Server Upgrade von HTTP 0.9 auf HTTP 1.0

Erweitern Sie Ihren HTTP 0.9 Server dahingehend, dass er HTTP 1.0 unterstützt (ohne Access Authentication, also ohne Security). Unter <https://www.w3.org/Protocols/HTTP/1.0/spec.html> finden Sie eine vollständige Übersicht der HTTP 1.0 Spezifikation. Bitte beachten Sie, dass keine Implementierung von Access Authentication erforderlich ist! Im Moodle finden Sie eine Zip-Datei „documentRoot.zip“ welche auch eine „form.html“ Datei beinhaltet. UM HTTP POST zu realisieren implementieren Sie das Parsen der erhaltenen Daten und einer Rückgabe dieser via HTML (siehe Beispielausgabe).

Der Server soll über Kommandozeilenparameter konfiguriert werden. Dabei müssen sowohl TCP-Port als auch Basisverzeichnis (document root) des HTTP-Servers beim Start angegeben werden.

Hinweise:

- Beachten Sie, dass HTTP 1.0 für diverse Fälle Statuscodes definiert. Behandeln Sie also auftretende Fehler dementsprechend und senden Sie den etwaigen Statuscode dem Client zurück.
- Sollte ein Verzeichnis angefordert werden, so soll der Server die in diesem Verzeichnis befindliche Datei index.html zurückgeben. Ist diese nicht vorhanden, so muss ein 404 Not Found zurück an den Client gesendet werden.
- Achten Sie beim zurück senden darauf dass die Header richtig gesetzt sind (wichtig ist hier die Felder „Content-Type“ und „Content-Length“).

Im Moodle-Kurs befinden sich ein paar Testdaten (documentRoot.zip), mit denen Sie Ihren Server mit einem Browser Ihrer Wahl testen können.

Beispielausgabe von Daten erhalten mittels eines Formulars und der POST Methode. Die Rückgabe der erhaltenen Daten erfolgt als HTML:

```
<html><body>
<p> Received form variable with name [firstname] and value [Mickey]</p>
<p> Received form variable with name [lastname] and value [Mouse]</p>
</body></html>
```

U 7.2 Java Sockets: HTTP-Server Upgrade von HTTP 1.0 auf HTTP 1.1 Persistent Connections

Erweitern Sie Ihren HTTP 1.0 Server dahingehend, dass er auch das Headerfeld „Connection“ unterstützt (weitere Informationen finden Sie hier: <https://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.10>). Hierzu müssen Sie persistente TCP Verbindungen implementieren, das heißt, dass die Verbindung zwischen Server und Client bestehen bleibt auch nach einem erfolgreichen Response (beachten Sie, dass dies von der angegebenen HTTP Version abhängt).