

	Datum	08.10.2021
Informatik		
Web Application Engineering	Übungsblatt 1	
Thomas Feilhauer	Zu lösen bis 15.10.2021, 8:00 Uhr	

Vorbereitungsaufgabe (in der Übungsstunde zu lösen):

Installieren Sie den Apache HTTP-Server

<http://httpd.apache.org/>

auf Ihrem Computer. Verwenden Sie dazu die Binary Distribution, die Sie unter

<https://www.apachelounge.com/download/>

oder

<https://www.apachehaus.com/cgi-bin/download.plx>

herunterladen können. Testen Sie den Webserver, indem Sie eine HTML-Datei ins Verzeichnis %**APACHE_HOME**%/htdocs einstellen, den HTTP-Server starten und dann in einem Web-Browser die entspr. Seite aufrufen; Sie können dafür eine beliebige Datei (z.B. HTML) aus dem Web kopieren. Falls sich der Webserver nicht auf Port 80 starten lässt, konfigurieren Sie ihn so, dass er auf einem anderen Port, z.B. Port 8080 startet.

Informationen über den Apache HTTP-Server finden Sie unter

<http://httpd.apache.org/docs/>

Machen Sie sich im Verlauf der kommenden Übungsaufgaben sukzessive mit den Einstellungen des Apache HTTP-Servers vertraut.

Aufgabe 1: Internet-Dienste

(2 Punkte)

Erklären Sie, was der Befehl

```
telnet time-A.timefreq.bldrdoc.gov 13
```

bewirkt. Welcher Dienst (der TCP/IP-Anwendungsschicht) wird aufgerufen?

Gemeint ist der Dienst, der durch obigen Befehl mit Hilfe von *telnet* adressiert wird.

Welcher Request for Comments (RFC) spezifiziert den Dienst? Was wird zurückgeliefert (ergänzen Sie Ihre Beschreibungen durch einen Screenshot)?

Hinweise für NutzerInnen von MS Windows:

- Telnet ist standardmäßig nicht freigeschaltet und muss erst bei den Windows-Features aktiviert werden, bevor es verwendet werden kann.
- Nach dem Aufruf von Telnet in der Windows-Console werden die Tastatureingaben nicht in der Windows-Console ausgegeben, sie werden aber sehrwohl als Eingabe entgegengenommen. D.h. die Tastatureingaben für Telnet müssen in der Windows-Console "blind" erfolgen. Die Ausgaben (Responses), die der verbundene Dienst zurückliefert, werden allerdings auf der Windows-Console angezeigt.
- In Windows-Console funktioniert CTRL-V nicht immer korrekt, um die Befehle z.B. aus einem Text-Editor heraus unter telnet in die Console zu pasten. Das Pasten mit der rechten Maustaste in die Console funktioniert hingegen einwandfrei!
- **Empfehlung:** Verwenden Sie das Windows-Subsystem für Linux oder verwenden Sie eine Virtuelle Maschine mit einem Linux-Betriebssystem. Beide Alternativen sind einfach zu installieren und können für verschiedenste Dienste nützlich sein – dort ist u.a. Telnet bereits aktiviert und alle Eingaben erscheinen am Terminal

Aufgabe 2: WebBrowser

(8 Punkte)

Implementieren Sie einen einfachen Web-Browser in Java. Dieser soll über eine Möglichkeit zur Eingabe einer URL verfügen, die die aufzurufende Web-Ressource referenziert. Die Textbestandteile der Entity aus der Response sollen bei einem Statuscode 200 als **Plain-Text** ausgegeben werden.

Funktionsweise des einfachen Web-Browsers:

- Erwarte die Eingabe einer URL durch den User.
- Teile die eingegebene URL auf in Domain-Name und Dokument-Referenz (URL-Path).
- Gib die Meldung „Connecting to Server <Domain name> at <IP-Adr.>“ aus.
- Baue eine TCP-Verbindung über einen Socket zu dem in der URL spezifizierten Host auf.
- Generiere einen entsprechenden HTTP-Request (HTTP 1.0 genügt) und sende ihn über die bestehende Verbindung.
- Warte auf die HTTP-Response-Message.
- Gib bei Statuscode 200 im Response nur die **Entity** aus der Response-Message aus (nicht den Header), ansonsten gib den Statuscode aus.

Für die Implementierung des Browsers dürfen **keine High-Level Funktionen** von Java eingesetzt werden, um Web-Dokumente zu laden (z.B.

`JEditorPane.setPage(String url)` oder alle Klassen in `java.net.URL*` sind tabu!). Die Kommunikation soll über explizit in Java programmierte TCP-Socket-Verbindungen erfolgen und die Generierung der HTTP-Requests ist im Programm zu codieren.

Der Browser kann als GUI-Anwendung mit einer entsprechenden scrollbaren Textausgabe und einem Send-Button oder als eine einfache Command-Line-Applikation implementiert werden. Es genügt, wenn der Web-Browser GET-Requests unterstützt und nur HTML-Dokumente anzeigt. Darüber hinaus kann der Zugriff auf die Default-Seite einer Domain eingeschränkt werden, d.h. ein URL-Path muss nicht ausgelesen werden.

Testen Sie Ihren Web-Browser für unterschiedliche URLs bzw. Webseiten, z.B. mit dem von Ihnen installierten HTTP-Server <http://localhost/>, aber nicht nur für die Default-Seiten, sondern auch für Unterseiten einer Webapplikation mit nicht-leerem URL-Pfad. Für welche Webseiten "funktioniert" Ihr Browser, für welche nicht? Dokumentieren Sie die URLs, mit denen Sie ihren Web-Browser getestet haben. Geben Sie zusätzlich zum Sourcecode ein ausführbares Jar-File für Ihre Applikation ab.