

Lernaufgabe

LE 3 – Web-Architektur und Web-basierte Kommunikation

Aufgabe 2: Einfacher Webserver

Lernziel

Sie sind in der Lage, die prinzipielle Funktionsweise eines Webserver und insbesondere die Kommunikation mittels HTTP zwischen einem Web-Client (z.B. Browser) und dem Webserver zu verstehen.

Aufgabe

Probieren Sie die den auf Moodle abgelegten einfachen in Java implementierten Webserver aus (LE 3, Aufgabe 2) und beantworten Sie folgende Fragen:

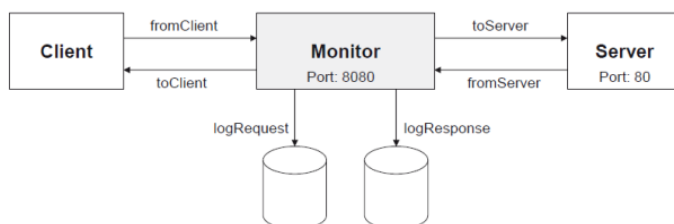
- Wie ist das Design des Webserver? Erstellen Sie dazu ein einfaches Klassendiagramm.
- Was unterstützt der Server vom HTTP-Protokoll und wie funktioniert das Programm grundsätzlich?
- Wie funktioniert genau das enthaltene Ajax-Beispiel (s. im Verzeichnis `/web` die Webseite `ajax.html`)?

Vorgehen

1. Entpacken Sie das ZIP-File in einem Arbeitsverzeichnis und importieren Sie das Projekt in Eclipse (Import->Existing Project into Workspace) oder einer anderen IDE.
2. Öffnen Sie für den Webserver ein Command- bzw. Shell-Prompt und wechseln Sie ins Verzeichnis des entpackten ZIP-Files.
3. Der Webserver wird aufgerufen mit: `java -cp bin MyWebserver` oder einfach mit dem beigelegten Skript `mywebserver` (Windows) bzw. `./mywebserver.sh` (MacOS, Linux).
4. Eine Test-Webseite `index.html` (im Verzeichnis `/web`) kann im Browser aufgerufen werden mit: `http://localhost:50000` oder `http://<ip address server>:50000`. Weiter hat es im `/web` Verzeichnis einfache Beispiele mit JavaScript (`javascript.html` und `ajax.html`).
5. Probieren Sie das Programm aus und schauen Sie sich den HTTP-Meldungsverkehr sowie den Source-Code an. Um den HTTP-Meldungsverkehr zu loggen, verwenden Sie das unter Hinweise beschriebene Programm oder eine andere dort beschriebene Variante.

Hinweise, Tipps

Das Programm Monitor (ebenfalls als ZIP-File auf Moodle) kann genutzt werden, um die zwischen Client und Server ausgetauschten Daten aufzuzeichnen. Aufruf im Verzeichnis des entpackten ZIP-Files: `java -cp bin Monitor <calling port> localhost <forwarding port> <request file name> <response file name>` oder einfach mit dem Skript `monitor`. Nach der Aufzeichnung das Programm mit `Ctrl + C` schliessen, um die Log-Dateien anzuschauen.



Das Programm Monitor fungiert hier als Proxy. Dieser leitet die Datenströme weiter an den Server (die Ausgabe des Client als Eingabe für den Server) bzw. an den Client (die Ausgabe des Servers als Eingabe für den Client). Anfragen und Antworten werden in getrennten Dateien protokolliert.



Der Meldungsverkehr kann je nach eingesetztem Browser auch direkt im Browser in einer Konsole (bei Chrome Einstellung unter „More Tools->Developer Tools“) angeschaut werden.

Die Datei `content-types.properties` im Verzeichnis des entpackten ZIP-Files enthält weitere MIME-Typen (z.B. für css), als die Java-Laufzeitumgebung standardmässig zur Verfügung stellt. Damit diese MIME-Typen verwendet werden können, muss der Web-Server wie folgt aufgerufen werden:

```
java -Dcontent.types.user.table=content-types.properties -cp bin MyWebserver. Im Skript wird der Webserver schon so aufgerufen.
```

Falls Sie die Programme ohne IDE (Eclipse o.a.) ändern und neu kompilieren möchten, können Sie dies mit dem Befehl `javac src*.java -d bin`. Noch einfacher geht es, wenn Sie Ant installiert haben (<http://ant.apache.org/>). Dann können Sie die Programme kompilieren indem Sie `ant` auf der Kommando-Zeile eingeben☺

Ergebnis

Fassen Sie die Erkenntnisse zur den Fragen zusammen und diskutieren Sie sie in Murmelgruppen.

Zeit: 30'