



Praktische Informatik 3: Funktionale Programmierung

Vorlesung 12 (17.01.2023): Übungen

Christoph Lüth



Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH



Universität
Bremen

Wintersemester 2022/23

Hands on!

Übung 12.1: HTTP handgemacht

Wenn nötig, installieren Sie das Programm telnet (oder schalten es, in Windows, frei). Starten Sie das Programm und verbinden Sie sich mit dem Web-Server der Uni Bremen:

```
telnet> open www.uni-bremen.de 80
```

Jetzt können Sie ein HTTP-Request von Hand eingeben (danach eine Leerzeile eingeben):

```
GET / HTTP/1.1  
Host: www.uni-bremen.de
```

Was passiert?

Der Server antwortet

Der Server antwortet

Lösung: Wahrscheinlich kommt die Antwort:

```
HTTP/1.1 301 Moved Permanently
Date: Sun, 31 Jan 2021 01:52:40 GMT
Server: Apache/2.4.29 (Ubuntu)
Location: https://www.uni-bremen.de/
Content-Length: 317
Content-Type: text/html; charset=iso-8859-1
```

gefolgt von etwas HTML. Damit teilt uns der Server mit, dass er nur noch https-Anfragen annimmt (und sagt uns, wo). Eventuell kommt auch

HTTP/1.1 400 Bad Request

...

dann war die Anfrage nicht HTTP-konform (wahrscheinlich vertippt?).

Erste Schritte.

Übung 12.2: Lecker Kekse!

Laden Sie die Quellen für die Vorlesung von heute herunter, und betreten Sie das Verzeichnis `simple-cookies`. Dort finden Sie ein einfaches Programm, welches mit Cookies zählt, wie oft eine Seite aufgerufen wurde, und das obligatorische Hello World, welches ein einfaches HTML-Dokument mittels Blaze erzeugt.

Erweitern Sie das Programm `SimpleCookies.hs` so dass es ein HTML-Dokument zurückgibt. Schreiben Sie dazu eine Funktion

```
hitPage :: String → Html
```

welche über die Anzahl der bisherigen Aufrufe (`hits'`) parametrisiert ist.

Eine mögliche Lösung

Eine mögliche Lösung

Lösung:

```
hitPage :: String → Html
hitPage hits =
    docTypeHtml $ do
        H.head $ do
            H.title "Simple-Cookies"
        H.body $ do
            H.h1 "Hello-World."
            H.span $ do
                H.text "You-have-been-here-"
                H.text (T.pack hits)
                H.text "-times."
```