Daniel Kröni

Arbeitsblatt: HTTP Web Request

Im Unterricht haben Sie gelernt wie man Konsolen-I/O programmiert und wie man Dateien lesen und schreiben kann. In diesem Arbeitsblatt lernen Sie, wie man aus einem Haskell Programm Internetdienste über HTTP (Hypertext Transfer Protocol) abfragen kann.

HTTP ist ein Protokoll zur Übertragung von Daten über ein Netzwerk. Es wird hauptsächlich eingesetzt um Webseiten aus dem Internet in einen Webbrowser zu laden. Die Kommunikation basiert dabei auf einem einfachen Anfrage-Antwort-Modell (Request-Response). Der Browser fordert eine Seite von einem Server an und bekommt als Antwort das HTML mit dem darzustellenden Inhalt.
Dieses Modell lässt sich aber nicht nur für HTML Internetseiten verwenden, sondern kann ganz allgemein verwendet werden um auf Netzwerkdienste zuzugreifen. Viele Internetdienstleister bieten Netzwerkschnittstellen an um programmatischen Zugriff auf die Dienste zu gewähren.

Unter der URL https://wttr.in/ können Wetterinfos im Text-Style abgefragt werden. Um das Wetter in Brugg im Kurzformat darzustellen, können Sie z.B. folgende URL verwenden:

`https://wttr.in/~Brugg?format=3`

Als Resultat kommt ein eine einzeilige Ausgabe:
`Brugg: ⛅  +17°C`

Sie können das im Browser ausprobieren.

In diesem Arbeitsblatt programmieren Sie nun ein Haskell Programm, um das Wetter an einem gewünschten Ort abzufragen.

Damit die Network.HTTP Libraries verfügbar sind, wurden diese als Abhängigkeit in der `WS-Webclient.cabal` Datei deklariert. Wechseln Sie in das Verzeichnis `WS_Webclient`. Folgende Kommandos sind nun nützlich:

Start von GHCi (wobei alle abhängigen Bibliotheken zur Verfügung stehen)
```
> cabal repl
```

Bauen und ausführen der `main` Funktion in der Datei `Main.hs` und Übergabe von einem Kommandozeilenargument (`arg0`)
```
> cabal run webclient arg0
```

Durch die Imports von `Network.HTTP.Client` und von `Network.HTTP.Client.TLS` stehen die notwendigen Funktionen zur Verfügung (die Typsignaturen sind vereinfacht dargestellt):

```
-- Ein I/O Programm, das ein https Manager liefert.
newManager tlsManagerSettings :: IO Manager


-- Nimmt eine URL (String) und erzeugt ein I/O Programm, das eine Anfrage liefert.
parseRequest :: String -> IO Request


-- Nimmt eine Anfrage und den https Manager und erzeugt ein I/O Programm,
-- das eine Antwort liefert.
httpLbs :: Request -> Manager -> IO (Response L8.ByteString)


-- Nimmt eine Antwort und liefert deren Hauptinhalt (in unserem Fall ein ByteString).
responseBody :: Response body -> body


-- Gibt einen ByteString auf der Konsole aus.
-- Verfügbar dank dem Import von Data.ByteString.Lazy.Char8.
L8.putStrLn :: L8.ByteString -> IO ()
```

Aufgabe:
Implementieren Sie ein Programm, das das Wetter von https://wttr.in/ lädt. Der Ort soll als Programmargument beim Aufruf mitgegeben werden können.