Stundenplan Parser in Haskell

Ein Projekt von Kevin Peters (70430220)

Inhalt

1. Motivation
2. Projekt
3. Umsetzung
4. Demo
5. Quellen

Motivation

- Stundenplanzugriff nur per Web

  ➜ HTTP, HTML

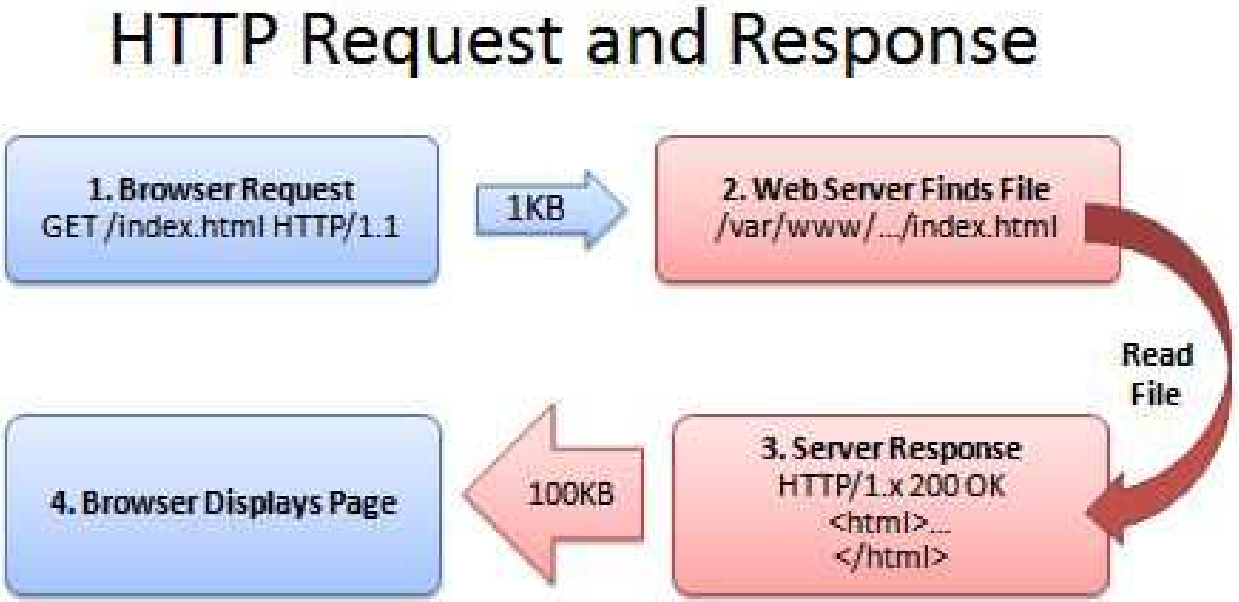
- HTML ist aufgebaut wie ein Baum

  ➜ rekursiver Ansatz

HTML Aufbau - Basic

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Title</title>
  <meta name="description" content="description">
  <meta name="author" content="Kevin Peters">
</head>
<body>
  <div>
    <h1>Heading</h1>
    <p>Paragraph</p>
  </div>
</body>
</html>
```

HTTP Request



HTTP - Sniffing

- Tool: Fiddler (http://www.telerik.com/fiddler) - Windows, Linux mit Mono
- Aufzeichnen der HTTP Requests und Responses

- Arten der Requests
  - GET
  - POST
  - PUT
  - DELETE
  - ...

Normaler Request

```
POST http://splus.ostfalia.de/semesterplan123.php?id=1362F014835FFFD0F67159E302EC1A3C&identifier=%23SPLUS659BF0
HTTP/1.1
Host: splus.ostfalia.de
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:50.0) Gecko/20100101 Firefox/50.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: de,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Referer:
http://splus.ostfalia.de/semesterplan123.php?id=1362F014835FFFD0F67159E302EC1A3C&identifier=%23SPLUS659BF0
Cookie: PHPSESSID=xxxxxxxxxxxxxxxxxxxxxxxxx
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 31

identifier%5B%5D=%23SPLUSB3BC2D
```

Anpassung des Requests

```
POST http://splus.ostfalia.de/semesterplan123.php?identifier=%23SPLUS659BF0 HTTP/1.1


identifier%5B%5D=%23SPLUSB3BC2D
```

HTTP Request in Kurzform

- POST
- id beschreibt den Kurs (Parameter)
- Header müssen nur teilweise gesetzt sein

➜ 302 mit Content wird als Response gegeben

- Eigentlich sollte das nicht passieren
- Einfachste Lösung

Id’s der Kurse

Id	**Vertiefungsrichtung/Jahr**
SPLUSB3BC20	B.Sc. - 2. Sem. Software Engineering (I-B-I2-SE)
SPLUS8677CD	B.Sc. - 4. Sem. Information Engineering (I-B-I4-IE)
SPLUS73FFC6	B.Sc. - 4. Sem. Medieninformatik (I-B-I4-MI)
SPLUS659BF0	B.Sc. - 4. Sem. Software Engineering (I-B-I4-SE)
SPLUS69C0D0	B.Sc. - 4. Sem. System Engineering (I-B-I4-SysE)

Darstellung der HTTP Request in Haskell

Libraries:

- wreq - http client

```
cabal update
cabal install -j --disable-tests wreq
```

- Control.Lens
  - Wird für Zugriff auf HTTP Response benötigt

HTTP Request

```
import Control.Lens
import Network.Wreq

standardHeader :: String
standardHeader = "identifier%5B%5D"

standardValue :: String
standardValue = "%23SPLUS659BF0"

postRequest id = do
    response <- post ("http://splus.ostfalia.de/semesterplan123.php?identifier=%23" ++ id) [(standardHeader) :=
(standardValue)]
    putStrLn "Done"
```

FEHLER!

```
[1 of 1] Compiling Main             ( httprequest.hs, interpreted )

httprequest.hs:11:94: error:
    * Couldn't match type `[Char]'
                    with `Data.ByteString.Internal.ByteString'
      Expected type: Data.ByteString.Internal.ByteString
        Actual type: String
    * In the first argument of `(:=)', namely `(standardHeader)'
      In the expression: (standardHeader) := (standardValue)
      In the second argument of `post', namely
        `[(standardHeader) := (standardValue)]'
Failed, modules loaded: none.
```

ByteString - String

- Zwei Varianten
  - Strict
  - Lazy
- 8-Bit String (Char8-Modul in Haskell vorhanden)
- Benutzung hier, da HTML nicht in UTF-8 encoded sein muss
- Bezug zu IO

Konvertierung String ➜ ByteString, ByteString ➜ String

```haskell
import qualified Data.ByteString.Lazy as B
import qualified Data.ByteString.Char8 as C

exampleString :: String
exampleString = "identifier%5B%5D"

exampleByteString = C.pack exampleString

{- To ByteString-}
toByteString input = C.pack input
{- To String -}
toString input = C.unpack (B.toStrict (input))
```

Probleme HTTP Response

- Durch Monaden Just/Maybe
- Eliminierung von Just/Maybe mithilfe von:
  - Data.Maybe
- Typkonvertierung von String

Beispiel für einen POST Request

```haskell
{-# LANGUAGE OverloadedStrings #-}
import Control.Lens
import Network.Wreq
import Data.Maybe
import qualified Data.ByteString.Lazy as B
import qualified Data.ByteString.Char8 as C

standardHeader :: String
standardHeader = "identifier%5B%5D"

standardValue :: String
standardValue = "%23SPLUS659BF0"

postRequest id = do
    response <- post ("http://splus.ostfalia.de/semesterplan123.php?identifier=%23" ++ id) [(C.pack standardHeader) := (C.pack standardValue)]
    let postResponseBody = response ^? responseBody
    let responseString = toString postResponseBody
    putStrLn responseString

toString input = C.unpack (B.toStrict (fromJust input))
```

HTML Aufbau

- Durch fehlerhaften Aufbau des HTML Dokuments, sehr schwierig zu parsen
- z.B. mehrere body-Tags (<body></body>)
- Gibt insgesamt 29 Fehler nach w3c validator

Tabellenstruktur - HTML

- <table>: Table
- <tr> : TableRow
- <td> : TableData
- <col> : column

Tabellenstruktur - Stundenplan

```
<tr>
  <td rowspan='1' class='row-label-one'>12:15</td>
  <td class='cell-border'> </td>
  <td class='cell-border'> </td>
  <td class='cell-border'> </td>
  <td class='cell-border'> </td>
  <td class='cell-border'> </td>
</tr>
<tr>
  <td rowspan='1' class='row-label-one'>12:30</td>
  <td class='cell-border'> </td>
  <td class='cell-border'> </td>
  <td class='cell-border'> </td>
  <td class='cell-border'> </td>
  <td class='cell-border'> </td>
</tr>
```

Tabellenstruktur - Stundenplan

```
<tr>
  <td rowspan='1' class='row-label-one'>12:00</td>
  <td class='cell-border'> </td>
  <td class='cell-border'> </td>
  <td class='cell-border'> </td>
  <td class='object-cell-border' colspan='1' rowspan='6'>
    <!-- START OBJECT-CELL -->
    <table class='object-cell-args' cellspacing='0' border='0' width='100%'>
      <col class='object-cell-0-1' />
      <tr>
        <td align='center'>MA-SoftwareEngineeringProjekt</td>
      </tr>
    </table>
    <table class='object-cell-args' cellspacing='0' border='0' width='100%'>
      <col class='object-cell-1-1' />
      <tr>
        <td align='center'></td>
      </tr>
    </table>
```

Tabellenstruktur - Stundenplan

```
    <table class='object-cell-args' cellspacing='0' border='0' width='100%'>
      <col class='object-cell-2-0' />
      <col class='object-cell-2-2' />
      <tr>
        <td align='left'>H&ouml;rsaal 127</td>
        <td align='right'>Prof. Dr. B. M&uuml;ller</td>
      </tr>
    </table>
    <!-- END OBJECT-CELL -->
  </td>
  <td class='cell-border'> </td>
  <td class='cell-border'> </td>
</tr>
```

Datenanalyse

Wir besitzen:

- Uhrzeit
- Raum
- Dozent
- Kurslänge (aus rowspan; 1 = 15 Minuten)
- Bestimmung des Wochentages?
- Bestimmung exaktes Datum?

Wochentag

- Zählen der <td>’s mit der Klasse ‘cell-border’
- Aktives Element ist <td> mit der Klasse ‘object-cell-border’

Aufkommen Nummer	**Wochentag**
1	Montag
2	Dienstag
3	Mittwoch
4	Donnerstag
5	Freitag

Parsing

- HTML Parser Libraries existieren bereits
- Selber geschrieben
  - Derzeit noch mit String-Operationen
  - Dynamische Struktur von HTML
- Eventueller Umstieg auf Parsing Library

Parsing

- Derzeit mit Data.List.Split
- Dabei werden Texte in Abschnitte getrennt
- Beispiel formatiert: <td align=’center’>Content</td>

➜ Content

```
getCourseName :: String -> String
getCourseName input = head (splitOn "</td>" (splitOn "<td align='center'>" input !! 1))
```

Parsing

- Dies wird für alle Elemente so ausgeführt
- HTML Encoding

➜ Library vorhanden: Web.Encodings

HTML	**Ascii**
ö	ö
ü	ü
Hörsaal 127	Hörsaal 127
Prof. Dr. B. Müller	Prof. Dr. B. Müller

Probleme

- Kurse, die nicht von Blockdauer sind (wie dieser hier gerade)
  - In der TableRow befindet sich ein Kurs
  - Verschiebung der Wochentag um -1 oder länger
  - Je nachdem wie lange die Kurse andauern wird verschoben
- Es kann auch sein, dass ein Kurs dort stattfindet, obwohl keine <td> ausgefüllt wurde
- Parallele Kurse zum gleichem Block

➜ Edge Case

Lösung

- Alle Kurse sammeln

  ➜ Danach durchgehen und verschieben

- Parser verfeinern

Demo

Beispiel

```
getSchedule "SPLUS659BF0" 51
[
((8,15),(9,45),"Thursday","IESE-IT-Sicherheit","H&ouml;rsaal 223","Prof. Dr. S. Gharaei"),
((9,0),(12,0),"Wednesday","SOE-Weitere Programmiersprache - Projektvortr&auml;ge","H&ouml;rsaal 223","Prof.
Dr. M. Huhn"),
((10,0),(11,30),"Wednesday","IESE-IT-Sicherheit","H&ouml;rsaal 223","Prof. Dr. S. Gharaei"),
((12,0),(13,30),"Thursday","SOE-Weitere Programmiersprache","H&ouml;rsaal 026","Prof. Dr. M. Huhn"),
((12,0),(13,30),"Friday","SOE-SeProjekt","Seminarraum 152","Prof. Dr. W. Pekrun"),
((14,15),(15,45),"Thursday","SOE-Weitere Programmiersprache","H&ouml;rsaal 026","Prof. Dr. M. Huhn"),
((14,15),(15,45),"Friday","SOE-SeProjekt","Seminarraum 152","Prof. Dr. W. Pekrun"),
((16,0),(17,30),"Wednesday","SOE-Fortgeschr. Themen Softwaretechnik","H&ouml;rsaal 026","Prof. Dr. B.
M&uuml;ller"),
((16,0),(17,30),"Thursday","SOE-Fortgeschr. Themen Softwaretechnik","H&ouml;rsaal 026","Prof. Dr. B.
M&uuml;ller")
]
```

Quellen

- http://www.html-seminar.de/bilder/doctype-aufbau-html5.png
- http://stackoverflow.com/questions/4109689/how-does-a-client-browser-generate-a-request-to-be-sent-to-a-server
- https://hackage.haskell.org/package/bytestring
- https://hackage.haskell.org/package/base-4.9.0.0/docs/Data-Maybe.html
- https://www.haskell.org/hoogle/
- https://hackage.haskell.org/package/wreq-0.4.1.0/docs/Network-Wreq.html
- http://www.serpentine.com/wreq/tutorial.html

Tools - Quellen

- http://markup.su/highlighter/
- https://www.google.com/slides/about/
- https://gist.github.com/