Лекция 5

HTTP – основни понятия

Въведение

- HTTP - акроним: HyperText Transfer Protocol
- протокол, използван от Уеб клиентите и сървърите да комуникират един с друг (да предават ресурси - HTML файлове, изображения, ...)).
- Браузера е Уеб клиент, тъй като създава (отваря) връзка към сървъра и изпраща *заявка (request message)* сървъра за да получи някакъв ресурс.
- Сървъра "обслужва" заявката и връща *отговор (response message)* - намира и връща заявения HTML файл или изпълнява заявената програма и връща резултата от нейното действие.

Формат на обменяните съобщения

Форматите на заявката и отговора са подобни. И двата вида съобщения съдържат:

- инициализиращ ред
- нула или повече заглавни (header) редове
- празен ред (CRLF)
- тяло (заявка, или отговор на заявката)

Или по-точно, формата на едно HTML съобщение е :

```
<инициализиращ ред>  # различава се при заявка или отговор
Header1: стойност1
Header1: стойност1
...
<празен ред>
<тяло >       # параметри на заявката или съдържание на файл
```

Инициализиращ ред при заявка

- *Initial Request Line* - състои се от три части, отделени чрез спейс: име на метод (на заявката), локален път до искания ресурс, използваната HTTP версия. Например:

  GET /path/to/file/index.html HTTP/1.0

  - GET - най-често използвания HTTP метод. Други са POST и HEAD. ! Винаги са с главни букви !
  - път - онази част от URL следваща името на хоста.
  - HTTP версия - винаги има формата: "HTTP/x.x"

Ако имаме заявен следния URL:

http://localhost/cgi-bin/hello2.cgi?name=Iva

то инициализиращия ред при тази заявка ще бъде:

GET /cgi-bin/hello2.cgi?name=Iva HTTP/1.1

Инициализиращ ред при отговор

- *Initial Response Line* (status line) - също се състои от три разделени чрез спейс части: HTTP версия, код за състояние (status code) на отговора, причина - пояснение на английски език. Например:

  HTTP/1.0 200 OK или HTTP/1.0 404 Not Found

- Статус кода е трицифрено число, като първата цифра определя категорията на съобщението:
  - 1xx — информативно съобщение
  - 2xx — успешно действие
  - 3xx — преадресация
  - 4xx — грешка от страна на клиента
  - 5xx — грешка от страна на сървъра

Заглавни редове (header lines)

Предоставят информация за заявката/отговора, за обекта в тялото на съобщението, ...

Всеки един заглавен ред има вида:

"**име на хедър : стойност**" , като редовете задължително се разполагат на отделни редове в началото.

Например:

User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.19) Gecko/20081217 K-Meleon/1.5.2

Content-Length: 580

If-Modified-Since: Sat, 29 Oct 1994 19:43:31 GMT

From: iva.e.popova@gmail.com

Тяло на съобщението

- При заявка: в тялото на съобщението се съдържат въведените от потребителя данни или файлове за качване
- при отговор: в тялото се съдържа ресурса, който е заявен от потребителя, или пък съобщението за грешка, ако такава е възникнала.
- Когато присъства тяло в съобщението, често в заглавните редове се съдържа информация за него. Например:

  `Content-Type:` дава MIME (Multipurpose Internet Mail Extensions) типа на данните в тялото: "text/html", "image/gif" и пр.

  `Content-Lenght:` броя байтове в тялото.

Пример за прост HTTP обмен

- **URL: http://127.0.0.1/Temp/hello.htm**
  1. Клиента се свързва с хоста: 127.0.0.1, порт 80
  2. Клиента изпраща следната заявка:

```
GET /Temp/hello.htm HTTP/1.1
Accept: image/gif,image/jpeg, image/png, */*
Connection: Keep-Alive
Host: 127.0.0.1
Accept-Language: en
User-Agent: Mozilla/4.0
```

  3. Сървъра търси заявения ресурс: **/Temp/hello.htm** и в случай че го намери връща следното съобщение:

```
HTTP/1.1 200 OK
Server: OptiPerl/Professional
Content-Type: text/html
Connection: close
Content-Length: 415

<HTML>
  # тук е съдържанието на страницата hello.html
</HTML>
```

  4. Сървъра прекратява връзката

Други HTTP методи за заявка: HEAD, POST

- HEAD - подобен на GET. Единствената разликата е че при HEAD заявките клиента изисква от сървъра само заглавните редове на ресурса, без да иска да получава съдържанието му. Отговора на HEAD заявка никога не съдържа тяло!
- POST - когато клиента изпраща данни към сървъра, които ще бъдат обработени чрез CGI/PHP и пр. приложение.

  Основните разлики между POST и GET са:

  - С POST винаги се предават данни, които се съдържат в самото тяло на заявката. Към заглавните редове се добавят допълнителни хедъри, като :Content-Type: и Content-Length:.

  - URI не е ресурс сам по себе си, а програма, която ще обработи получените данни.

  - HTTP отговора на такива заявки е изхода от съответната програма, а не статичен документ.

Предаване на данни към сървъра

- Най-често, когато се предават данни чрез POST заявка, в хедъра на клиентското съобщение се добавя редовете:

  Content-Type : application/x-www-form-urlencoded

  Content-Length : 15

  ! Когато за предаване на данните се използва GET, Content-Type е същия, (макар и да не се указва явно) както и процедурата по кодирането на съобщението (url encoding)

- Когато данните се предават чрез GET то те стават достъпни за CGI приложението чрез променлива на средата QUERY_STRING (в Пърл: $ENV{QUERY_STRING})

- Когато данните се предават чрез POST, то те се подават към приложението посредством стандартния поток за вход (STDIN) -- приложението трябва да се погрижи да прочете необходимия брой байтове (указани в Content-length хедъра).

Литература

on-line:

- http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html
  - w3.org дефиниции на статус кода

on-store:

- Клинтън Уонг, *HTTP - джобен справочник*, СофтПрес, 2001, http://www.bgbook.dir.bg/book.php?ID=7772