HTTP

Ненко Табаков
Пламен Танов
Технологическо училище “Електронни системи”
Технически университет – София

4 ноември 2008

ЛИТЕРАТУРА НЕОБХОДИМИ ПРОГРАМИ

- SUN's Java Sockets Tutorial - http://java.sun.com/docs/books/tutorial/networking/sockets/
- Java API документация - http://java.sun.com/javase/6/docs/api/
- HTTP Turotial - http://www.jmarshall.com/easy/http/
- Eclipse - www.eclipse.org
- Apache HTTP Client - http://hc.apache.org/httpclient-3.x/

СЪДЪРЖАНИЕ

- Какво е ***HTTP***
  - Какво е ресурс
- Структура на ***HTTP*** транзакция
  - Начална заявка
  - Начален отговор
  - Хедър
  - Тяло на съобщение
  - ***HEAD*** метод
  - ***POST*** метод
- ***HTTP 1.1*** изисквания

КАКВО Е ***HTTP***

- ***HTTP – Hypertext Transfer Protocol***
- Мрежови протокол, който се използва за пренос на данни през ***World Wide Web***
- Един браузър е ***HTTP*** клиент, който праща заявки и приема отговори от ***HTTP*** сървър
- Портът, на който един ***HTTP*** сървър слуша, по подразбиране, е 80, но може да се ползва който и да е

КАКВО Е РЕСУРС

- ***HTTP*** пренася ресурси
- Ресурс е парче информация, която може да се идентифицира с ***URL***
- Един ресурс може да бъде както файл, така и само текст или част от някакъв документ

СТРУКТУРА НА ***HTTP*** ТРАНЗАКЦИЯ

- ***HTTP***, като повечето мрежови протоколи, използва клиент – сървър модел
- ***HTTP*** клиент отваря връзка и изпраща заявка към ***HTTP*** сървър
- Сървърът от своя страна отговаря, като обикновено в отговора му се съдържа ресурс
- След доставяне на отговора връзката между клиента и сървъра се затваря

СТРУКТУРА НА *HTTP* ТРАНЗАКЦИЯ

•Формата на заявката и отговора са подобни и изглеждат по следния начин:

- Начална линия
- Нула или повече хедъри
- Празна линия
- Съобщение, което не е задължително

ПРИМЕР СТРУКТУРА НА *HTTP* ТРАНЗАКЦИЯ

```
<начална линия, различна в зависимост дали е заявка или
отговор>
Header1: value1
Header2: value2
Header3: value3

<тук се разполага съобщение, което не е задължително. Може да
бъде дълго много линии и дори да съдържа двоични данни>

GET /path/file.html HTTP/1.0
From: someuser@jmarshall.com
User-Agent: HTTPTool/1.0
```

НАЧАЛНА ЛИНИЯ НА ЗАЯВКА

- Началната линия на заявка се състои от три части:
  - Име на метода
  - Локален път до искания ресурс
  - Версия на ***HTTP***, която се използва

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

НАЧАЛНА ЛИНИЯ НА ЗАЯВКА

- Освен ***GET*** има и други методи – ***POST,HEAD,DELETE, PUT*** и други
- Методите винаги са с главни букви
- Пътят е ***URL,*** след името на хоста
- Версията винаги има синтаксис - ***HTTP/x.x***

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

НАЧАЛНА ЛИНИЯ НА ОТГОВОР

•Началната линия на отговора (статус линия) също се състои от три части:

- Версия на ***HTTP***, която се използва
- Статус код на отговора
- Фраза описващата статус кода

```
HTTP/1.0 200 OK

HTTP/1.0 404 Not Found
```

НАЧАЛНА ЛИНИЯ НА ОТГОВОР

- Версията винаги има синтаксис – ***HTTP/x.x***
- Статус кодът е предназначен за четене от компютър, а фразата, описваща го е предназначена за четене от хора
- Статус кодът се състои от три цифри
- Първата цифра идентифицира категорията на отговора:
  - **1xx** – идентифицира информационно съобщение
  - **2xx** – идентифицира успех
  - **3xx** – пренасочва клиента към друго ***URL***
  - **4xx** – идентифицира грешка от страна на клиента
  - **5xx** – идентифицира грешка от страна на сървъра

```
HTTP/1.0 200 OK

HTTP/1.0 404 Not Found
```

НАЧАЛНА ЛИНИЯ НА ОТГОВОР

- Най – често срещаните статус кодове са:
  - **200 OK –** заявката е успешна и изискания ресурса се връща в часта на съобщението
  - **404 Not Found –** Изискваният ресурс не съществува
  - **301 Moved Permanently**
  - **302 Moved Temporarily**
  - **303 See Other –** Ресурсът е преместен на друг ***URL*** и би трябвало автоматично да се изтегли от клиента
  - **500 Server Error –** Неочаквана грешка от страна на сървъра

ХЕДЪРИ

- Хедърите предоставят информация за заявката/отговора или относно съдържанието на съобщението
- Синтаксисът на хедърите е следният:
  - Един хедър е един ред
  - Тя има следната форма: ***Име: стойност***
  - Името на хедъра може да бъде както с малки, така и с големи букви и те са еднакви (не е case sensitive)

```
Следващите два хедъра са еквивалентни:

Content-Type: text/html
CONTENT-TYPE: text/html
```

ХЕДЪРИ

- ***HTTP 1.0*** дефинира 16 хедъра и нито един от тях не е задължителен
- ***HTTP 1.1*** дефинира 46 хедъра и един (***Host:***) е задължителен
- ***From:*** - дава e-mail адреса на този, който прави заявката
- ***User-Agent***: - идентифицира програмата, която прави заявката. Формата му е ***ИмеНаПрограмата/x.x***
- ***Server:*** - идентифицира софтуера на сървъра. Има същата форма като горепоказаната
- ***Last-Modified*** – дава дата на модифициране на ресура в следния формат Last-Modified: Fri, 31 Dec 1999 23:59:59 GMT

ТЯЛО НА СЪОБЩЕНИЕТО

- Всяка заявка или отговор може да съдържа съобщение след хедърите
- В отговора това е мястото, където се съдържа искания ресурс
- В заявката това е мястото, където потребителят е въвел данни или се съдържа файла за качване на сървъра
- Ако има съобщение обикнове се срещат следните хедъри:
  - ***Content-Type*** – дава ***MIME***-типа на данните в съобщението (***text/html, image/gif***)
  - ***Content-Length*** **–** дава броя на байтовете в тялото на съобщението

ПРИМЕР

За да свалим съдържанието на http://www.somehost.com/path/file.html е нужно:

- Да отворим сокет към www.somehost.com на порт 80
- Да изпратим следната ***HTTP*** заявка

```
GET /path/file.html HTTP/1.0
From: someuser@jmarshall.com
User-Agent: HTTPTool/1.0

```

ПРИМЕР

От своя страна сървърът ще отговори с нещо подобно на:

```
HTTP/1.0 200 OK
Date: Fri, 31 Dec 1999 23:59:59 GMT
Content-Type: text/html
Content-Length: 1354

<html>
<body>
<h1>Happy New Millennium!</h1>
(more file contents)
   .
   .
   .
</body>
</html>
```

HEAD МЕТОДЪТ

- ***HEAD*** заявка е същата като ***GET*** заявка с разликата, че казва на сървъра да върне само хедърите без ресурс.
- Това е удобен метод, когато е нужно да се знаят характеристиките на един ресурс, без да е нужно сваляне то му
- По този начин браузърите могат да осъществяват кеширане на уеб страници

POST МЕТОДЪТ

- ***POST*** заявка се използва при пращане на данни към сървъра, които трябва да бъдат обработени по някакъв начин
- ***POST*** заявка се различава от ***GET*** заявка по следните неща:
  - Обикновено има хедъри, които описват съдържанието на съобщението (***Content-Type, Content-Length***)
  - Пътят не е адрес към ресурс за сваляне, а програма, която да обработи данните, които се пращат
  - Една такава завка предполага промяна на ресурс от сървъра
  - Отговорът не е статичен файл, а изхода на програмата, която е обработила данните

ПРИМЕР

```
POST /path/script.cgi HTTP/1.0
From: frog@jmarshall.com
User-Agent: HTTPTool/1.0
Content-Type: application/x-www-form-urlencoded
Content-Length: 32

home=Cosby&favorite+flavor=flies
```

HTTP 1.1 ИЗИСКВАНИЯ

- Един сървър (един IP адрес) може да съдържа няколко домайна
- Например ***www.host1.com*** и ***www.host2.com*** може да се помещават на един сървър (едно IP)
- Поради тази причина хоста трябва да бъде изрично упоменат и евентуално порта, на който работи
- Това става с хедъра ***Host:***

```
GET /path/file.html HTTP/1.1
Host: www.host1.com:80
```