

HTTP
Hypertext Transfer Protocol

Refs: RFC 1945 (HTTP 1.0)

Netprog 2001 HTTP

1

HTTP Usage

- HTTP is the protocol that supports communication between web browsers and web servers.
- A "Web Server" is a HTTP server
- We will look at HTTP Version 1.0

Netprog 2001 HTTP

2

From the RFC

"HTTP is an application-level protocol with the lightness and speed necessary for distributed, hypermedia information systems."

Netprog 2001 HTTP

3

Transport Independence

- The RFC states that the HTTP protocol generally takes place over a TCP connection, but the HTTP protocol itself is not dependent on a specific transport layer.

Netprog 2001 HTTP

4

Request - Response

- HTTP has a simple structure:
 - client sends a request
 - server returns a reply.
- HTTP can support multiple request-reply exchanges over a single TCP connection, but this is a special case.

Netprog 2001 HTTP

5

Well Known Address

- The “well known” TCP port for HTTP servers is port 80.
- Other ports can be used as well...

Netprog 2001 HTTP

6

HTTP Versions

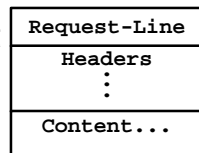
- The original version now goes by the name "HTTP Version 0.9"
 - HTTP 0.9 was used for many years.
- Starting with HTTP 1.0 the version number is part of every request.
- HTTP is still changing...

Netprog 2001 HTTP

7

HTTP 1.0 Request

- Lines of text (ASCII).
- Lines end with CRLF "`\r\n`"
- First line is called "Request-Line"



Netprog 2001 HTTP

8

Request Line

Method URI HTTP-Version \r\n

- The request line contains 3 *tokens* (words).
- space characters " " separate the tokens.
- Newline seems to work by itself (but the protocol requires CRLF)

Netprog 2001 HTTP

9

Request Method

- The Request Method can be:

GET HEAD PUT
POST DELETE LINK
UNLINK

future expansion allowed

Netprog 2001 HTTP

10

Methods

- GET: retrieve information identified by the URI.
- HEAD: retrieve meta-information about the URI.
- POST: send information to a URI and retrieve result.

Netprog 2001 HTTP

11

Methods (cont.)

- PUT: Store information in location named by URI.
- DELETE: remove *entity* identified by URI.
- LINK, UNLINK: create/destroy a link relationship...?...?

Netprog 2001 HTTP

12

Common Usage

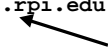
- GET, HEAD and POST are supported everywhere.
- HTTP 1.1 servers often support PUT, DELETE, OPTIONS & TRACE.

Netprog 2001 HTTP

13

URI

Universal Resource Identifier

- URIs defined in RFC 1630.
- Full URI: `proto://hostname/path`
`http://www.cs.rpi.edu:80/blah/foo`
 Identifies the Server
- Partial URI: `/path`
`/blah/foo`

No server mentioned

Netprog 2001 HTTP

14

URI Usage

- When dealing with a HTTP server, only a partial URI is used.
- When dealing with a *proxy* HTTP server, a full URI is used.
 - client has to tell the proxy where to get the document!
 - *more on proxy servers in a bit...*

Netprog 2001 HTTP

15

HTTP Version Number

“HTTP/1.0” OR “HTTP/1.1”

HTTP 0.9 did not include a version number in a request line.

If a server gets a request line with no HTTP version number it assume 0.9

Netprog 2001 HTTP

16

The Header Lines

- After the *Request-Line* come a number of HTTP *headers*.
- Each header line contains an attribute name followed by a “:” followed by the attribute value.

Netprog 2001 HTTP

17

Headers

- Request Headers provide information to the server about the client
 - what kind of client
 - what kind of content will be accepted
 - who is making the request
- There can be 0 headers!

Netprog 2001 HTTP

18

Example HTTP Headers

```
Accept: text/html
From: neytmann@cybersurg.com
User-Agent: Netscape 4.7
Referer: http://foo.com/blah
```

Netprog 2001 HTTP

19

End of the Headers

- Each header ends with a CRLF
- The end of the header section is marked with a blank line
- `"\r\n\r\n"`
- For GET and HEAD requests the end of the headers is the end of the request!

Netprog 2001 HTTP

20

POST

- A POST request includes some *data* after the headers (after the blank line).
- There is no format for the data (just raw bytes).
- A POST request must include a Content-Length line in the headers:
`Content-Length: 267`

Netprog 2001 HTTP

21

Example Request

```
GET /~hollingd/testanswers.html HTTP/1.0
Accept: */*
User-Agent: Internet Explorer
From: cheater@cheaters.org
Referer: http://foo.com/
```

Netprog 2001 HTTP

22

Example Post

```
POST /CGI-BIN/add_appointments HTTP/1.0
Accept: */*
User-Agent: Internet Explorer
Content-Length: 34

1220=surgery&0110=doom&0320=bypass
```

Netprog 2001 HTTP

23

Typical Method Usage

GET used to retrieve an HTML document.

HEAD used to find out if a document has changed.

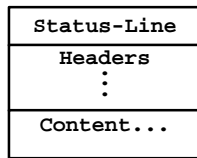
POST used to submit a form.

Netprog 2001 HTTP

24

HTTP Response

- ASCII Status Line
- Headers Section
- Content can be anything (not just text)
 - typically is HTML document



Netprog 2001 HTTP

25

Response Status Line

HTTP-Version Status-Code Message

- *Status Code* is 3 digit number (for computers)
- *Message* is text (for humans)

Netprog 2001 HTTP

26

Status Codes

- 1xx Informational
- 2xx Success
- 3xx Redirection
- 4xx Client Error
- 5xx Server Error

Netprog 2001 HTTP

27

Example Status Lines

`HTTP/1.0 200 OK`

`HTTP/1.0 301 Moved Permanently`

`HTTP/1.0 400 Bad Request`

`HTTP/1.0 500 Internal Server Error`

Netprog 2001 HTTP

28

Response Headers

- Provide the client with information about the returned *entity* (document).
 - what kind of document
 - how big the document is
 - how the document is encoded
 - when the document was last modified
- Response headers end with blank line

Netprog 2001 HTTP

29

Response Header Examples

`Date: Thu, 27 Jan 2000 12:48:17 EST`

`Server: Apache/1.17`

`Content-Type: text/html`

`Content-Length: 1756`

`Content-Encoding: gzip`

Netprog 2001 HTTP

30

Content

- Content can be anything (sequence of raw bytes).
- Content-Length header is required for any response that includes content.
- Content-Type header also required

Netprog 2001 HTTP

31

Try it with telnet

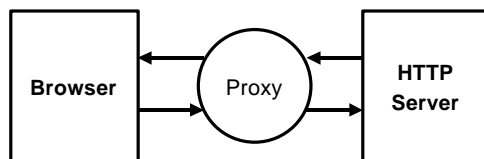
```
> telnet www.cs.rpi.edu 80  
GET / HTTP/1.0  
HTTP/1.0 200 OK  
Server: Apache  
...
```

Request (points to GET / HTTP/1.0)
Blank Line (end of headers) (points to blank line between GET and HTTP/1.0)
Response (points to Server: Apache)

Netprog 2001 HTTP

32

HTTP Proxy Server



Netprog 2001 HTTP

33
