Advanced Programming in the UNIX Environment

Week 10: HTTP

Department of Computer Science
Stevens Institute of Technology

Jan Schaumann
jschauma@stevens.edu
https://stevens.netmeister.org/631/
HTTP

The Hypertext Transfer Protocol — A simple request/response protocol.

- RFC1945 (HTTP/1.0; May 1996)
- RFC2616 (HTTP/1.1; June 1999)
- RFC7540 (HTTP/2; May 2015)
- RFC9114 (HTTP/3; June 2022)
The Hypertext Transfer Protocol — A simple request/response protocol.

- RFC1945 (HTTP/1.0; May 1996)
- RFC2616 (HTTP/1.1; June 1999)
- RFC7540 (HTTP/2; May 2015)
- RFC9114 (HTTP/3; June 2022)
The Hypertext Transfer Protocol

HTTP is a request/response protocol:
1. client sends a request to the server
2. server responds
The Hypertext Transfer Protocol

HTTP is a request/response protocol:

1. client sends a request to the server

   $ telnet www.google.com 80
   Trying 2607:f8b0:4006:809::2004...
   Connected to www.google.com.
   Escape character is '^['].
The Hypertext Transfer Protocol

HTTP is a request/response protocol:

1. client sends a request to the server
   - request method

   $ telnet www.google.com 80
   Trying 2607:f8b0:4006:809::2004...
   Connected to www.google.com.
   Escape character is '^]'.
   **GET**
HTTP is a request/response protocol:

1. Client sends a request to the server
   - request method
   - URI

   $ telnet www.google.com 80
   Trying 2607:f8b0:4006:809::2004...
   Connected to www.google.com.
   Escape character is '^]'.
   GET /
The Hypertext Transfer Protocol

HTTP is a request/response protocol:

1. client sends a request to the server
   • request method
   • URI
   • protocol version

$ telnet www.google.com 80
Trying 2607:f8b0:4006:809::2004...
Connected to www.google.com.
Escape character is '^]'.
GET / HTTP/1.0
The Hypertext Transfer Protocol

HTTP is a request/response protocol:

1. client sends a request to the server
   • request method
   • URI
   • protocol version
   • [request modifiers]

$ telnet www.google.com 80
Trying 2607:f8b0:4006:809::2004...
Connected to www.google.com.
Escape character is '^[']'.
GET / HTTP/1.0
If-Modified-Since: Mon, 13 Nov 2023 18:03:39 GMT
The Hypertext Transfer Protocol

HTTP is a request/response protocol:

1. client sends a request to the server
   • request method
   • URI
   • protocol version
   • [request modifiers]
   • [client information]

   $ telnet www.google.com 80
   Trying 2607:f8b0:4006:809::2004...
   Connected to www.google.com.
   Escape character is '^]'.
   GET / HTTP/1.0
   If-Modified-Since: Mon, 13 Nov 2023 18:03:39 GMT
   User-Agent: Mozilla/5.0
The Hypertext Transfer Protocol

HTTP is a request/response protocol:

1. client sends a request to the server
2. server responds

• status line (including success or error code)

```
GET / HTTP/1.0
If-Modified-Since: Mon, 13 Nov 2023 18:03:39 GMT
User-Agent: Mozilla/5.0

HTTP/1.0 200 OK
Date: Mon, 13 Nov 2023 19:21:10 GMT
Server: gws
Content-Type: text/html; charset=ISO-8859-1

<!doctype html><html itemscope=""
itemtype="http://schema.org/WebPage"
lang="en"><head><meta
```
The Hypertext Transfer Protocol

HTTP is a request/response protocol:

1. client sends a request to the server
2. server responds
   • status line (including success or error code)
   • server information
   • entity meta-information
   • content

GET / HTTP/1.0
If-Modified-Since: Mon, 13 Nov 2023 18:03:39 GMT
User-Agent: Mozilla/5.0

HTTP/1.0 200 OK
Date: Mon, 13 Nov 2023 19:21:10 GMT
Server: gws
Content-Type: text/html; charset=ISO-8859-1

<!doctype html><html itemscope=""
itemtype="http://schema.org/WebPage"
lang="en"><head><meta
The Hypertext Transfer Protocol

HTTP is a request/response protocol:
1. client sends a request to the server
2. server responds
   • status line (including success or error code)
   • server information
   • entity meta-information
   • content

GET / HTTP/1.0
If-Modified-Since: Mon, 13 Nov 2023 18:03:39 GMT
User-Agent: Mozilla/5.0
HTTP/1.0 200 OK
Date: Mon, 13 Nov 2023 19:21:10 GMT
Server: gws
Content-Type: text/html; charset=ISO-8859-1

<!doctype html><html itemscope=""
itemtype="http://schema.org/WebPage"
lang="en"><head><meta
The Hypertext Transfer Protocol

HTTP is a request/response protocol:

1. client sends a request to the server
2. server responds
   - status line (including success or error code)
   - server information
   - entity meta-information
   - content

```
GET / HTTP/1.0
If-Modified-Since: Mon, 13 Nov 2023 18:03:39 GMT
User-Agent: Mozilla/5.0

HTTP/1.0 200 OK
Date: Mon, 13 Nov 2023 19:21:10 GMT
Server: gws
Content-Type: text/html; charset=ISO-8859-1
<!doctype html><html itemscope=""
itemtype="http://schema.org/WebPage"
lang="en"><head><meta
```
The Hypertext Transfer Protocol

Server status codes:

• 1xx – Informational; Request received, continuing process
• 2xx – Success; The action was successfully received, understood, and accepted
• 3xx – Redirection; Further action must be taken in order to complete the request
• 4xx – Client Error; The request contains bad syntax or cannot be fulfilled
• 5xx – Server Error; The server failed to fulfill an apparently valid request
The Hypertext Transfer Protocol

$ telnet www.stevens.edu 80
[...]
GET / HTTP/1.1
Host: www.stevens.edu

HTTP/1.1 301 Moved Permanently
Date: Mon, 13 Nov 2023 20:10:40 GMT
Transfer-Encoding: chunked
Connection: keep-alive
Cache-Control: max-age=3600
Expires: Mon, 13 Nov 2023 21:10:40 GMT
Location: https://www.stevens.edu/
Server: cloudflare
The Hypertext Transfer Protocol

$ openssl s_client -crlf -servername www.stevens.edu -connect www.stevens.edu:443

[...]  
GET / HTTP/1.1
Host: www.stevens.edu

HTTP/1.1 200 OK
Date: Mon, 13 Nov 2023 20:12:48 GMT
Content-Type: text/html; charset=utf-8
Transfer-Encoding: chunked
Connection: keep-alive
Cache-Control: public, max-age=0, must-revalidate
Strict-Transport-Security: max-age=15552000
The Hypertext Transfer Protocol

HTTP is a *Transfer* Protocol – serving data, not any specific text format.

- *Accept* or *Accept-Encoding* client header can specify different formats (e.g., `application/json`) or encodings such as *gzip*, *Brotli* etc.
- corresponding server headers: *Content-Type* and *Content-Encoding*

Data can be transferred in either direction.

- *POST*, *PUT*, *DELETE*
- *GET*, *HEAD*
- *CONNECT*, *OPTIONS*
HTTP - more than just static data

HTTP is a Transfer Protocol – what is transferred need not be static; resources may generate different data to return based on many variables.

- CGI – resource is executed, needs to generate appropriate response headers
- server-side scripting (ASP, PHP, Perl, …)
- client-side scripting (JavaScript/ECMAScript/JScript,…)
- applications based on HTTP, using:
  - AJAX
  - RESTful services
  - JSON, XML, YAML to represent state and abstract information
HTTP in your final project

• protocol versions supported: 1.0
• request methods supported: GET, HEAD
• request headers supported: If-Modified-Since
• response headers included: Date, Server, Last-Modified, Content-Type, Content-Length
HTTP in your final project

Let’s create your building blocks:

• program startup and daemonization
• rough network logic
• input / request parsing and validation
• cgi handling
• user directory handling
• directory index generation
• regular file serving
Links

- https://www.w3.org/Protocols/
- https://stevens.netmeister.org/631/f23-group-project.html
- https://www.netmeister.org/cgi-bin/env.cgi