

Hypertext Transport Protocol



HTTP



- Hypertext Transport Protocol
- Language of the Web
 - protocol used for communication between web browsers and web servers
- TCP port 80

HTTP - URLs



- URL

- Uniform Resource Locator

- protocol (http, ftp, news)
 - host name (name.domain name)
 - port (usually 80 but many on 8080)
 - directory path to the resource
 - resource name

- `http://xxx.myplace.com/www/index.html`

- `http://xxx.myplace.com:80/cgi-bin/t.exe`

HTTP - methods

- **Methods**

- **GET**

- retrieve a URL from the server
 - simple page request
 - run a CGI program
 - run a CGI with arguments attached to the URL

- **POST**

- preferred method for forms processing
 - run a CGI program
 - parameterized data in sysin
 - more secure and private

HTTP - methods

- Methods (cont.)
 - PUT
 - Used to transfer a file from the client to the server
 - HEAD
 - requests URLs status header only
 - used for conditional URL handling for performance enhancement schemes
 - retrieve URL only if not in local cache or date is more recent than cached copy

HTTP Request Packets



- Sent from client to server
- Consists of HTTP header
 - header is hidden in browser environment
 - contains:
 - content type / mime type
 - content length
 - user agent - browser issuing request
 - content types user agent can handle
- and a URL

HTTP Request Headers



- Precede HTTP Method requests
- headers are terminated by a blank line
- Header Fields:
 - From
 - Accept
 - Accept-Encoding
 - Accept Language

HTTP Request Headers (cont.)



- Referer
- Authorization
- Charge-To
- If-Modified-Since
- Pragma

From:



- In internet mail format, the requesting user
- Does not have to correspond to requesting host name (might be a proxy)
- should be a valid e-mail address

Accept:




- List of schemes which will be accepted by client
- `<field> = Accept: <entry> * [,<entry>]`
- `<entry> = <content type> *[:,<param>]`
- `<param> = <attr> = <float>`
- `<attr> = q / mxs / mxb`
- `<float> = <ANSI-C floating point >`
- `Accept: text/html`
- `Accept: audio/basic q=1`
- if no Accept is found; plain/text is assumed
- may contain wildcards (*)

Accept-Encoding



- Like Accept but list is a list of acceptable encoding schemes
- Ex
 - Accept-Encoding: x-compress;x-zip

User-Agent



- Software product used by original client
- `<field> = User-Agent: <product>`
- `<product> = <word> [/<version>]`
- `<version> = <word>`
- Ex.
 - User-Agent: IBM WebExplorer DLL /v960311

Referer



- For Server's benefit, client lists URL of document (or document type) from which the URL in request was obtained.
- Allows server to generate back-links, logging, tracing of bad links...
- Ex.
 - Referer: <http://www.w3.com/xxx.html>

Authorization:



- For Password and authentication schemes
- Ex.
 - Authorization: user fred:mypassword
 - Authorization: kerberos kerberosparameters

ChargeTo:



- Accounting information
- Accounting system dependent

Pragma:



- Same format as accept
- for servers
- should be passed through proxies, but used by proxy
- only pragma currently defined is no-cache; proxy should get document from owning server rather than cache

Modified-Since:



- Used with GET to make a conditional GET
- if requested document has not been modified since specified date a Modified 304 header is sent back to client instead of document
 - client can then display cached version

Response Packets



- Sent by server to client browser in response to a Request Packet

Status Header



- “HTTP/1.0 sp code”
- Codes:
 - 1xx - reserved for future use
 - 2xx - successful, understood and accepted
 - 3xx - further action needed to complete
 - 4xx - bad syntax in client request
 - 5xx - server can't fulfill good request

HTTP Response Headers

- Sent by server to client browser
- Status Header
 - Entities
 - Content-Encoding:
 - Content-Length:
 - Content-Type:
 - Expires:
 - Last-Modified:
 - extension-header
- Body – content (usually html)

Status Codes



- 200 OK
- 201 created
- 202 accepted
- 204 no content
- 301 moved perm.
- 302 moved temp
- 304 not modified
- 400 bad request
- 401 unauthorized
- 403 forbidden
- 404 not found
- 500 int. server error
- 501 not impl.
- 502 bad gateway
- 503 svc not avail

Statelessness



- Because of the Connect, Request, Response, Disconnect nature of HTTP it is said to be a stateless protocol
 - i.e. from one web page to the next there is nothing in the protocol that allows a web program to maintain program “state” (like a desktop program).
 - “state” can be maintained by “witchery” or “trickery” if it is needed

Maintaining program “state”



- Hidden variables (`<input type=hidden>`)
- Sessions
 - Special header tags interpreted by the server
 - Used by ASP, PHP, JSP
 - Implemented at the language api level