

HTTP 2

Recall

Most of the HTTP material is on the previous HTTP handout

Request

Response -- code/head/body

MIME type

telnet 80 -- see response

Use the "telnet www.stanford.edu 80" to see the server response

dumpenv.pl -- see request

The following special program running on www-cs-faculty will print out what your browser is sending as its request. It's linked off the course page.

<http://www-cs-faculty.stanford.edu/cgi-bin/nick/dumpenv.pl>

Here's what it looks like when I click the link from my office
(nick3.stanford.edu)...

```
SERVER_SOFTWARE = Apache/1.3.0 (Unix)
GATEWAY_INTERFACE = CGI/1.1
DOCUMENT_ROOT = /u/www/htdocs          ## the doc-root in the file-sys
REMOTE_ADDR = 171.64.64.167
SERVER_PROTOCOL = HTTP/1.0
REQUEST_METHOD = GET
REMOTE_HOST = nick3.stanford.edu      ## where my browser is running
HTTP_REFERER = http://www.stanford.edu/class/cs193i/  ## page URL came from
QUERY_STRING =
HTTP_USER_AGENT = Mozilla/4.77 (Macintosh; U; PPC)    ## my browser
PATH = /sbin:/usr/sbin:/usr/bin
TZ = US/Pacific
HTTP_CONNECTION = Keep-Alive
HTTP_ACCEPT = image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, image/png,
*/*
REMOTE_PORT = 1838
HTTP_ACCEPT_LANGUAGE = en
SCRIPT_NAME = /cgi-bin/nick/dumpenv.pl
SCRIPT_FILENAME = /u/httpd/cgi-bin/nick/dumpenv.pl
HTTP_ACCEPT_ENCODING = gzip
SERVER_NAME = Sunburn.Stanford.EDU
REQUEST_URI = /cgi-bin/nick/dumpenv.pl          ## what came after the "GET"
HTTP_ACCEPT_CHARSET = iso-8859-1,*,utf-8
HTTP_COOKIE = SUNetCookieBrowser=TRUE;
SITESERVER=ID=3af7a299albe81a423146cc31ff92448
SERVER_PORT = 80
HTTP_HOST = www-cs-faculty.stanford.edu
SERVER_ADMIN = webmaster@cs.stanford.edu
```

Request HEAD

Use instead of GET -- returns just the header and omits the body.

Use to get the header without the expense of transmitting the whole body.

Some servers do not implement the HEAD, but they are in error; it's part of the standard.

Opening in a new Window

To cause a link to open in new window, add a target="_blank" binding in the anchor tag :. This works if the browser supports frames.

Empty Request

http://www.yahoo.com -- path,file, suffix all omitted

In this case -- we'll append a "/" as the path "http://www.yahoo.com" ->

"http://www.yahoo.com/". All the browsers do this. This behavior is now an official special case in the HTTP standard.

Relative URL Examples

"Base url" - the full url of the enclosing page e.g. http://foo.com/a/b.html"

1. Relative URL (i.e. not starting with /) picks up scheme,host,path from base URL: "c.html" -> "http://foo.com/a/c.html"
2. Root relative URL, picks up scheme and host from base url: "/images/smile.jpeg" -> "http://foo.com/images/smile.jpeg"

"index.html"

Server convention

Most HTTP servers, such as Apache, can be set so that given a request for a directory, such as /class/cs193i/, the server will search for a file with a special name such as "index.html" in that directory. If the file is there, it is returned.

If there is no index.html, the server may generate a directory listing on the fly or generate a 404.

Not HTTP protocol

This behavior is an option on the server side, it is not part of the HTTP protocol.

Servers, such as Apache, are highly configurable -- they have many settings for how the deal with various server-side situations. The default file may be "index.html" or "default.html", or "index.htm" for lame operating systems.

Don't use "dir/index.html" in URLs -- in HTML or given to people

It's simpler to just always use the "dir/" form. This way, there is only one "name" to refer to that one document. Otherwise, you can get multiple URLs that really point to the same thing. It can confuse the web robots, and it can make the "visited url" coloring wrong in the browser.

Yes: /usr/class/cs193i/

No: /usr/class/cs193i/index.html

/ Redirect -- 300

e.g. /class/cs193i -- no trailing slash
directory / convention

Most servers use the convention that a path ending a slash identifies a directory, otherwise a file.

No File

There is no file named /class/cs193i, but there is a directory -- the server figures that's what the client meant

302 Moved

Server answers back with a 302 (or 301, 303, ...) error message basically saying the resource is not available. BUT...

Location: /class/cs193i/ -- redirect

Server includes a location: url field in the header that directs the client to a new url to use. This is a server redirect.

Client Side

All the major browsers automatically re-try the request with the new URL, so the user doesn't notice. However, the "corrected" url is now the one that shows up at the top of the window.

1. Simple Uses

Fix client naming goof-ups

2. Complex Uses

The client clicks on a link. The server wants to send them to a different page dynamically depending on -- who they are, what server is managing their session, etc.. Custom server programming can use a dynamically computed redirect to bounce the client to a different server with much more flexibility than a fixed url.

Changing URL

You can tell a redirect is happening when you type URL in, but suddenly your browser re-writes it to be a different URL because it got a 300 response from the server.

Suppose did not correct client

Suppose the 300 error were not used, and instead the server added in / where necessary and sent back the HTML

1. Suppose client requests "/foo" where foo is really a directory
2. The server figures the client meant "/foo/" and so sends back the contents of "/foo/index.html"
3. There's an "a.html" relative URL in the HTML. The client does the rel->abs conversion and gets "/a.html", when the correct URL was "/foo/a.html"

Lesson: the client and the server need to have a consistent understanding of where the "/" are in the URLs. That's why the 300 errors are required to correct the client.