Web Server Design

Lecture 2 – URIs, Logs, MIME

Old Dominion University

Department of Computer Science

CS 431/531 Spring 2026

David Calano <dcala005@odu.edu>

Original slides by Michael L. Nelson

so about that `Host:` request header…

(a parable about software vs. specifications)

The `Host:` Request Header

```
$ telnet bit.ly 80
Trying 67.199.248.11...
Connected to bit.ly.
Escape character is '^]'.
HEAD http://bit.ly/2ogMITK HTTP/1.1
Connection: close

HTTP/1.1 400 Bad Request
Server: nginx
Date: Wed, 05 Sep 2018 03:25:20 GMT
Content-Type: text/html
Content-Length: 166
Connection: close

Connection closed by foreign host.
```

```
$ telnet bit.ly 80
Trying 67.199.248.11...
Connected to bit.ly.
Escape character is '^]'.
HEAD http://bit.ly/2ogMITK HTTP/1.1
Host: foo.bar.edu
Connection: close

HTTP/1.1 301 Moved Permanently
Server: nginx
Date: Wed, 05 Sep 2018 03:26:40 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 161
Connection: close
Cache-Control: private, max-age=90
Location: http://ws-dl.blogspot.com/2016/10/
 2016-10-13-dodging-memory-hole-2016.html

Connection closed by foreign host.
```

RFC 7230: (5.4 Host)

A client MUST send a Host header field in all HTTP/1.1 request messages. If the target URI includes an authority component, then a client MUST send a field-value for Host that is identical to that authority component, excluding any userinfo subcomponent and its "@" delimiter (Section 2.7.1). If the authority component is missing or undefined for the target URI, then a client MUST send a Host header field with an empty field-value.

…

A server MUST respond with a 400 (Bad Request) status code to any HTTP/1.1 request message that lacks a Host header field and to any request message that contains more than one Host header field or a Host header field with an invalid field-value.

another look at bit.ly

```
$ telnet bit.ly 80
Trying 67.199.248.11...
Connected to bit.ly.
Escape character is '^]'.
HEAD http://bit.ly/2ogMITK HTTP/1.1
Host: foo.bar.edu
Host: foo2.bar.edu
Host: really.should.send.a.400
Connection: close

HTTP/1.1 301 Moved Permanently
Server: nginx
Date: Wed, 05 Sep 2018 13:51:37 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 161
Connection: close
Cache-Control: private, max-age=90
Location: http://ws-dl.blogspot.com/2016/10/
 2016-10-13-dodging-memory-hole-2016.html
```

```
$ telnet bit.ly 80
Trying 67.199.248.10...
Connected to bit.ly.
Escape character is '^]'.
HEAD http://bit.ly/2ogMITK HTTP/1.1
Host: sldjflasjdljdfl
Connection: close

HTTP/1.1 301 Moved Permanently
Server: nginx
Date: Wed, 05 Sep 2018 13:52:13 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 161
Connection: close
Cache-Control: private, max-age=90
Location: http://ws-dl.blogspot.com/2016/10/
 2016-10-13-dodging-memory-hole-2016.html
```

RFCs 7230—7235
have replaced RFC 2616

but

2616 is what most software still implements

RFC 2616: (5.2 The Resource Identified by a Request)

The exact resource identified by an Internet request is determined by examining both the Request-URI and the Host header field.

An origin server that does not allow resources to differ by the requested host MAY ignore the Host header field value when determining the resource identified by an HTTP/1.1 request. (But see section 19.6.1.1 for other requirements on Host support in HTTP/1.1.)

An origin server that does differentiate resources based on the host requested (sometimes referred to as virtual hosts or vanity host names) MUST use the following rules for determining the requested resource on an HTTP/1.1 request:

1. If Request-URI is an absoluteURI, the host is part of the Request-URI. Any Host header field value in the request MUST be ignored.

2. If the Request-URI is not an absoluteURI, and the request includes a Host header field, the host is determined by the Host header field value.

3. If the host as determined by rule 1 or 2 is not a valid host on the server, the response MUST be a 400 (Bad Request) error message.

```
$ telnet www.cs.odu.edu 80
Trying 128.82.4.2...
Connected to xenon.cs.odu.edu.
Escape character is '^]'.
GET /~mln/index.html HTTP/1.1
Connection: close
Host: foo.bar.edu

HTTP/1.1 200 OK
Date: Mon, 23 Jan 2006 01:59:19 GMT
Server: Apache/1.3.26 (Unix) ApacheJServ/1.1.2 PHP/4.3.4
Last-Modified: Sun, 29 May 2005 02:46:53 GMT
ETag: "1c52-14ed-42992d1d"
Accept-Ranges: bytes
Content-Length: 5357
Connection: close
Content-Type: text/html

[deletia]
```

Is This RFC 2616 Compliant?

```
$ telnet www.cs.odu.edu 80
Trying 128.82.4.2...
Connected to xenon.cs.odu.edu.
Escape character is '^]'.
HEAD http://lajsdflakjsdlj.aslkdfjldsj.foo.com/~mln/index.htmlHTTP/1.1
Host: www.cs.odu.edu
Connection: close

HTTP/1.1 200 OK
Date: Fri, 30 Jan 2009 21:17:46 GMT
Server: Apache/2.2.0
Last-Modified: Wed, 14 Jan 2009 16:45:46 GMT
ETag: "88849-1cfe-1247a280"
Accept-Ranges: bytes
Content-Length: 7422
Connection: close
Content-Type: text/html
```

Is This RFC 2616 Compliant?

```
$ telnet www.google.com 80
Trying 209.85.165.99...
Connected to www.l.google.com.
Escape character is '^]'.
HEAD / HTTP/1.1
Connection: close

HTTP/1.1 200 OK
Cache-Control: private
Content-Type: text/html
Set-Cookie:
PREF=ID=d9086367498004ae:TM=1169488419:LM=1169488419:S=L0vxDxm20siPrfQi;
expires=Sun, 17-Jan-2038 19:14:07 GMT; path=/; domain=.google.com
Server: GWS/2.1
Content-Length: 0
Date: Mon, 22 Jan 2007 17:53:39 GMT

Connection closed by foreign host.
```

Different, but probably still not compliant (404 vs. 400)

```
$ telnet www.google.com 80
Trying 172.217.14.68...
Connected to www.google.com.
Escape character is '^]'.
HEAD http://lajsdflakjsdlj.aslkdfjldsj.foo.com/~mln/index.html HTTP/1.1
Host: www.cs.odu.edu
Connection: close

HTTP/1.1 404 Not Found
Content-Type: text/html; charset=UTF-8
Referrer-Policy: no-referrer
Content-Length: 1576
Date: Wed, 05 Sep 2018 03:40:03 GMT
Connection: close
```

This is 2616/7230 Compliant

```
$ telnet www.cs.odu.edu 80
Trying 128.82.4.2...
Connected to xenon.cs.odu.edu.
Escape character is '^]'.
HEAD / HTTP/1.1
Connection: close

HTTP/1.1 400 Bad Request
Date: Mon, 22 Jan 2007 17:56:07 GMT
Server: Apache/2.2.0
Connection: close
Content-Type: text/html;
charset=iso-8859-1

Connection closed by foreign host.
```

This is RFC 1945 compliant!

```
$ telnet bit.ly 80
Trying 67.199.248.10...
Connected to bit.ly.
Escape character is '^]'.
HEAD http://bit.ly/2ogMITK HTTP/1.0
Connection: close

HTTP/1.1 301 Moved Permanently
Server: nginx
Date: Wed, 05 Sep 2018 13:57:59 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 161
Connection: close
Cache-Control: private, max-age=90
Location: http://ws-dl.blogspot.com/2016/10/
 2016-10-13-dodging-memory-hole-2016.html

Connection closed by foreign host.
```

```
$ openssl s_client -connect www.cs.odu.edu:443
[ssl deletia]
HEAD /~mln/index.html HTTP/1.1
Host: foo.bar.edu

HTTP/1.1 200 OK
Server: nginx
Date: Wed, 05 Sep 2018 04:10:02 GMT
Content-Type: text/html
Connection: keep-alive
Vary: Accept-Encoding
Front-End-Https: on

HEAD http://www.cs.odu.edu/~mln/index.html HTTP/1.1
Connection: close
Host: foo.bar.edu

HTTP/1.1 200 OK
Server: nginx
Date: Wed, 05 Sep 2018 04:10:06 GMT
Content-Type: text/html
Connection: close
Vary: Accept-Encoding
Front-End-Https: on

closed
```

It does not appear that servers’ processing matches 5.2 of 2616, (and certainly not 5.4 of 7230)

Nice cleanup of “Identifying a Target Resource” in RFC 7230

(cf. 5.2 in 2616)

5.3. Request Target

Once an inbound connection is obtained, the client sends an HTTP request message (Section 3) with a request-target derived from the target URI. There are four distinct formats for the request-target, depending on both the method being requested and whether the request is to a proxy.

```
request-target = origin-form
               / absolute-form
               / authority-form
               / asterisk-form
```

(cf. 5.1.2 of 2616)

5.3.1. origin-form

The most common form of request-target is the origin-form.

implied preference not present in 2616!

```
origin-form    = absolute-path [ "?" query ]
```

When making a request directly to an origin server, other than a CONNECT or server-wide OPTIONS request (as detailed below), a client MUST send only the absolute path and query components of the target URI as the request-target. If the target URI's path component is empty, the client MUST send "/" as the path within the origin-form of request-target. A Host header field is also sent, as defined in Section 5.4.

For example, a client wishing to retrieve a representation of the resource identified as

```
http://www.example.org/where?q=now
```

directly from the origin server would open (or reuse) a TCP connection to port 80 of the host "www.example.org" and send the lines:

```
GET /where?q=now HTTP/1.1
Host: www.example.org
```

followed by the remainder of the request message.

5.3.2. absolute-form

When making a request to a proxy, other than a CONNECT or server-wide OPTIONS request (as detailed below), a client MUST send the target URI in absolute-form as the request-target.

```
absolute-form  = absolute-URI
```

The proxy is requested to either service that request from a valid cache, if possible, or make the same request on the client's behalf to either the next inbound proxy server or directly to the origin server indicated by the request-target. Requirements on such "forwarding" of messages are defined in Section 5.7.

An example absolute-form of request-line would be:

```
GET http://www.example.org/pub/WWW/TheProject.html HTTP/1.1
```

To allow for transition to the absolute-form for all requests in some future version of HTTP, a server MUST accept the absolute-form in requests, even though HTTP/1.1 clients will only send them in requests to proxies.

5.3.3. authority-form

The authority-form of request-target is only used for CONNECT requests (Section 4.3.6 of [RFC7231]).

```
  authority-form = authority
```

When making a CONNECT request to establish a tunnel through one or more proxies, a client MUST send only the target URI's authority component (excluding any userinfo and its "@" delimiter) as the request-target. For example,

```
  CONNECT www.example.com:80 HTTP/1.1
```

5.3.4. asterisk-form

The asterisk-form of request-target is only used for a server-wide OPTIONS request (Section 4.3.7 of [RFC7231]).

```
asterisk-form  = "*"
```

When a client wishes to request OPTIONS for the server as a whole, as opposed to a specific named resource of that server, the client MUST send only "*" (%x2A) as the request-target. For example,

```
OPTIONS * HTTP/1.1
```

If a proxy receives an OPTIONS request with an absolute-form of request-target in which the URI has an empty path and no query component, then the last proxy on the request chain MUST send a request-target of "*" when it forwards the request to the indicated origin server.

For example, the request

```
OPTIONS http://www.example.org:8001 HTTP/1.1
```

would be forwarded by the final proxy as

```
OPTIONS * HTTP/1.1
Host: www.example.org:8001
```

after connecting to port 8001 of host "www.example.org".

RFC 7230:
2.7.1. http URI Scheme

Note that the presence of a URI with a given authority component does not imply that there is always an HTTP server listening for connections on that host and port. Anyone can mint a URI. What the authority component determines is who has the right to respond authoritatively to requests that target the identified resource. The delegated nature of registered names and IP addresses creates a federated namespace, based on control over the indicated host and port, whether or not an HTTP server is present.

this is important: you can use http URIs to name things, but it doesn’t mean that there will always be a representation for that resource

Identifiers and http URIs

```
VIN
6 A 63 Q 217870
| | |  | |
| | |  | |- consecutive unit number
| | |  |--- 8 cylinder 428 cid (4V) (428 cubic inches ~= 7 litres)
| | |------ 7 Litre 2 door convertible
| |-------- assembled in Atlanta
|---------- 1966 model
```

Q: how to distinguish “6A63Q217870” as a VIN vs. some random hash value?

(one possible) A: promote to an http URI:

http://vin.ford.com/6A63Q217870
http://6A63Q217870.fords.mln/
http://michaelssupervindecoder.com/?vin=6A63Q217870

These URIs identify my convertible, but the authorities either don’t exist or are outside of my control. Techniques for asserting that all these URIs identify the same convertible is out of scope for this class.

Origin form vs. Absolute form

Previously known as: Absolute & Relative URIs

```
$ telnet www.cs.odu.edu 80 | tee 3-1.out
Trying 128.82.4.2...
Connected to xenon.cs.odu.edu.
Escape character is '^]'.
GET /~mln/index.html HTTP/1.1
Connection: close
Host: www.cs.odu.edu

[deletia]

$ telnet www.cs.odu.edu 80 | tee 3-2.out
Trying 128.82.4.2...
Connected to xenon.cs.odu.edu.
Escape character is '^]'.
GET http://www.cs.odu.edu/~mln/index.html HTTP/1.1
Connection: close
Host: www.cs.odu.edu

[deletia]

$ diff 3-1.out 3-2.out
5c5
< Date: Mon, 23 Jan 2006 01:54:49 GMT
---
> Date: Mon, 23 Jan 2006 01:55:02 GMT
```

2.7.1. http URI Scheme

The "http" URI scheme is hereby defined for the purpose of minting identifiers according to their association with the hierarchical namespace governed by a potential HTTP origin server listening for TCP ([RFC0793]) connections on a given port.

```
http-URI = "http:" "//" authority path-abempty [ "?" query ] [ "#" fragment ]
```

The origin server for an "http" URI is identified by the authority component, which includes a host identifier and optional TCP port ([RFC3986], Section 3.2.2). The hierarchical path component and optional query component serve as an identifier for a potential target resource within that origin server's name space. The optional fragment component allows for indirect identification of a secondary resource, independent of the URI scheme, as defined in Section 3.5 of [RFC3986].

A sender MUST NOT generate an "http" URI with an empty host identifier. A recipient that processes such a URI reference MUST reject it as invalid.

If the host identifier is provided as an IP address, the origin server is the listener (if any) on the indicated TCP port at that IP address. If host is a registered name, the registered name is an indirect identifier for use with a name resolution service, such as DNS, to find an address for that origin server. If the port subcomponent is empty or not given, TCP port 80 (the reserved port for WWW services) is the default.

2.7.2. https URI Scheme

The "https" URI scheme is hereby defined for the purpose of minting identifiers according to their association with the hierarchical namespace governed by a potential HTTP origin server listening to a given TCP port for TLS-secured connections ([RFC5246]).

All of the requirements listed above for the "http" scheme are also requirements for the "https" scheme, except that TCP port 443 is the default if the port subcomponent is empty or not given, and the user agent MUST ensure that its connection to the origin server is secured through the use of strong encryption, end-to-end, prior to sending the first HTTP request.

```
  https-URI = "https:" "//" authority path-abempty [ "?" query ] [ "#" fragment ]
```

Note that the "https" URI scheme depends on both TLS and TCP for establishing authority. Resources made available via the "https" scheme have no shared identity with the "http" scheme even if their resource identifiers indicate the same authority (the same host listening to the same TCP port). They are distinct namespaces and are considered to be distinct origin servers. However, an extension to HTTP that is defined to apply to entire host domains, such as the Cookie protocol [RFC6265], can allow information set by one service to impact communication with other services within a matching group of host domains.

```
$ telnet www.cs.odu.edu 80
Trying 128.82.4.2...
Connected to xenon.cs.odu.edu.
Escape character is '^]'.
HEAD /~mln/index.html HTTP/1.1
Host: www.cs.odu.edu
Connection: close

HTTP/1.1 301 Moved Permanently
Server: nginx
Date: Wed, 05 Sep 2018 16:00:58 GMT
Content-Type: text/html
Connection: close
Location: https://www.cs.odu.edu/~mln/index.html

Connection closed by foreign host.
```

Listen to the server…
It is telling you...
This resource has a new URI

In practice, people canonicalize http and https URIs to the same value, and while that is almost always true, it is not guaranteed

http and https URI Normalization and Comparison (RFC 7230)

2.7.3. http and https URI Normalization and Comparison

Since the "http" and "https" schemes conform to the URI generic syntax, such URIs are normalized and compared according to the algorithm defined in Section 6 of [RFC3986], using the defaults described above for each scheme.

If the port is equal to the default port for a scheme, the normal form is to omit the port subcomponent. When not being used in absolute form as the request target of an OPTIONS request, an empty path component is equivalent to an absolute path of "/", so the normal form is to provide a path of "/" instead. The scheme and host are case-insensitive and normally provided in lowercase; all other components are compared in a case-sensitive manner. Characters other than those in the "reserved" set are equivalent to their percent-encoded octets: the normal form is to not encode them (see Sections 2.1 and 2.2 of [RFC3986]).

For example, the following three URIs are equivalent:

http://example.com:80/~smith/home.html
http://EXAMPLE.com/%7Esmith/home.html
http://EXAMPLE.com:/%7esmith/home.html

AKA URL canonicalization: https://github.com/iipc/urlcanon

2.2. Reserved Characters

RFC 3986

URIs include components and subcomponents that are delimited by characters in the "reserved" set. These characters are called "reserved" because they may (or may not) be defined as delimiters by the generic syntax, by each scheme-specific syntax, or by the implementation-specific syntax of a URI's dereferencing algorithm. If data for a URI component would conflict with a reserved character's purpose as a delimiter, then the conflicting data must be percent-encoded before the URI is formed.

```
reserved    = gen-delims / sub-delims

   gen-delims  = ":" / "/" / "?" / "#" / "[" / "]" / "@"

   sub-delims  = "!" / "$" / "&" / "'" / "(" / ")" / "*" / "+" / "," / ";" / "="
```

The purpose of reserved characters is to provide a set of delimiting characters that are distinguishable from other data within a URI. URIs that differ in the replacement of a reserved character with its corresponding percent-encoded octet are not equivalent. Percent-encoding a reserved character, or decoding a percent-encoded octet that corresponds to a reserved character, will change how the URI is interpreted by most applications. Thus, characters in the reserved set are protected from normalization and are therefore safe to be used by scheme-specific and producer-specific algorithms for delimiting data subcomponents within a URI.

2.3. Unreserved Characters

RFC 3986

Characters that are allowed in a URI but do not have a reserved purpose are called unreserved. These include uppercase and lowercase letters, decimal digits, hyphen, period, underscore, and tilde.

```
unreserved  = ALPHA / DIGIT / "-" / "." / "_" / "~”
```

URIs that differ in the replacement of an unreserved character with its corresponding percent-encoded US-ASCII octet are equivalent: they identify the same resource. However, URI comparison implementations do not always perform normalization prior to comparison (see Section 6). For consistency, percent-encoded octets in the ranges of ALPHA (%41-%5A and %61-%7A), DIGIT (%30-%39), hyphen (%2D), period (%2E), underscore (%5F), or tilde (%7E) should not be created by URI producers and, when found in a URI, should be decoded to their corresponding unreserved characters by URI normalizers.

2.4. When to Encode or Decode

RFC 3986

Under normal circumstances, the only time when octets within a URI are percent-encoded is during the process of producing the URI from its component parts. This is when an implementation determines which of the reserved characters are to be used as subcomponent delimiters and which can be safely used as data. Once produced, a URI is always in its percent-encoded form.

When a URI is dereferenced, the components and subcomponents significant to the scheme-specific dereferencing process (if any) must be parsed and separated before the percent-encoded octets within those components can be safely decoded, as otherwise the data may be mistaken for component delimiters. The only exception is for percent-encoded octets corresponding to characters in the unreserved set, which can be decoded at any time. For example, the octet corresponding to the tilde ("~") character is often encoded as "%7E" by older URI processing implementations; the "%7E" can be replaced by "~" without changing its interpretation.

Because the percent ("%") character serves as the indicator for percent-encoded octets, it must be percent-encoded as "%25" for that octet to be used as data within a URI. Implementations must not percent-encode or decode the same string more than once, as decoding an already decoded string might lead to misinterpreting a percent data octet as the beginning of a percent-encoding, or vice versa in the case of percent-encoding an already percent-encoded string.

Encoding is often skipped

http://web.archive.org/web/19971015020026/http://www.larc.nasa.gov:80/

http://web.archive.org/web/19971015020053/http://www.larc.nasa.gov:80/larc.cgi?+s1#s1

https://via.hypothes.is/https://www.bartleby.com/101/549.html

https://donotlink.it/https://www.cs.odu.edu/~mln/

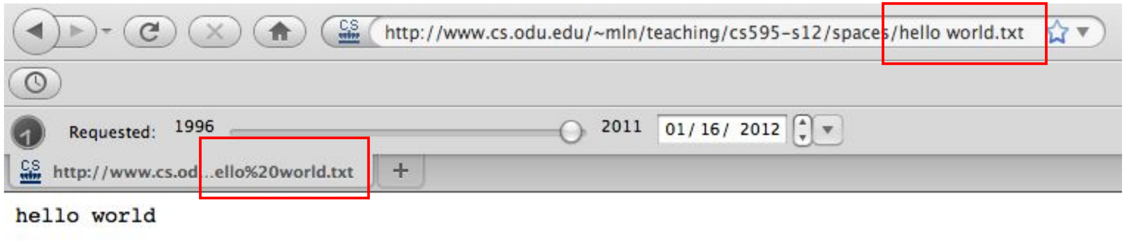
URIs that are essentially APIs establish positional arguments,
which allow for manual editing of URIs without the encumbrance of encoding

Encoding Cheat Sheet

Character	URI Role	Escape Sequence
/	Path Component Separator	%2F
?	Query Component Separator	%3F
#	Fragment Identifier	%23
=	Name/Value Separator	%3D
&	Argument Separator in Query Component	%26
:	Host Port Separator	%3A
;	Authority Namespace Separator	%3B
	Space Character	%20
%	Escape Indicator	%25
+	Escaped Space	%2B

From: http://www.openarchives.org/OAI/openarchivesprotocol.html

Clients Hide Details!



http://www.cs.odu.edu/~mln/teaching/cs595-s12/spaces/hello world.txt
Requested:
1996
2011
01/16/ 2012
http://www.cs.od...ello%20world.txt

hello world

```
$ telnet www.cs.odu.edu 80
Trying 128.82.4.2...
Connected to xenon.cs.odu.edu.
Escape character is '^]'.
HEAD http://www.cs.odu.edu/~mln/teaching/cs595-s12/spaces/hello world.txt HTTP/1.1
Host: www.cs.odu.edu
Connection: close

HTTP/1.1 404 Not Found
Date: Tue, 17 Jan 2012 02:27:17 GMT
Server: Apache/2.2.17 (Unix) PHP/5.3.5 mod_ssl/2.2.17 OpenSSL/0.9.8q
Connection: close
Content-Type: text/html; charset=iso-8859-1
```

Spaces Are Encoded

```
$ telnet www.cs.odu.edu 80
Trying 128.82.4.2...
Connected to xenon.cs.odu.edu.
Escape character is '^]'.
HEAD http://www.cs.odu.edu/~mln/teaching/cs595-s12/spaces/hello%20world.txt
HTTP/1.1
Host: www.cs.odu.edu
Connection: close

HTTP/1.1 200 OK
Date: Tue, 17 Jan 2012 02:27:54 GMT
Server: Apache/2.2.17 (Unix) PHP/5.3.5 mod_ssl/2.2.17 OpenSSL/0.9.8q
Last-Modified: Tue, 17 Jan 2012 02:25:55 GMT
ETag: "c-4b6b00cf932d2"
Accept-Ranges: bytes
Content-Length: 12
Connection: close
Content-Type: text/plain
```

Fragments (from RFC 1630)

This represents a part of, fragment of, or a sub-function within, an object. Its syntax and semantics are defined by the application responsible for the object, or the specification of the content type of the object.

…

The fragment-id follows the URL of the whole object from which it is separated by a hash sign (#). If the fragment-id is void, the hash sign may be omitted: A void fragment-id with or without the hash sign means that the URL refers to the whole object.

…

A reference to a particular part of a document may, including the fragment identifier, look like

http://www.myu.edu/org/admin/people#andy

in which case the string "#andy" is not sent to the server, but is retained by the client and used when the whole object had been retrieved.

Stated less clearly in section 4.1 of RFC 2396
also in 3.5 of RFC 3986

URLs & Filenames?

HIERARCHICAL FORMS

> The slash ("/", ASCII 2F hex) character is reserved for the delimiting of substrings whose relationship is hierarchical. This enables partial forms of the URI. Substrings consisting of single or double dots ("." or "..") are similarly reserved.
>
> The significance of the slash between two segments is that the segment of the path to the left is more significant than the segment of the path to the right. ("Significance" in this case refers solely to closeness to the root of the hierarchical structure and makes no value judgement!)

Note

> The similarity to unix and other disk operating system filename conventions should be taken as purely coincidental, and should not be taken to indicate that URIs should be interpreted as file names.

RFC 1630; restated in various ways in other RFCs

TRACE Method

```
$ telnet www.cs.vt.edu 80
Trying 198.82.184.180...
Connected to cs.vt.edu.
Escape character is '^]'.
TRACE / HTTP/1.1
Testing-1: Hello, Sailor!
Testing-2: 69,105
Host: www.cs.odu.edu
Connection: close

HTTP/1.1 200 OK
Date: Mon, 16 Jan 2012 23:23:32 GMT
Server: Apache/2.2.3 (CentOS)
Connection: close
Transfer-Encoding: chunked
Content-Type: message/http

77
TRACE / HTTP/1.1
Testing-1: Hello, Sailor!
Testing-2: 69,105
Host: www.cs.odu.edu.edu.edu.edu
Connection: close


0
```

ignore for now

TRACE no longer on www.cs.odu.edu

```
$ telnet www.cs.odu.edu 80
Trying 128.82.4.2...
Connected to xenon.cs.odu.edu.
Escape character is '^]'.
TRACE / HTTP/1.1
Testing-1: Hello, Sailor!
Testing-2: 69,105
Host: www.cs.odu.edu
Connection: close

HTTP/1.1 405 Method Not Allowed
Date: Mon, 16 Jan 2012 23:23:14 GMT
Server: Apache/2.2.17 (Unix) PHP/5.3.5 mod_ssl/2.2.17 OpenSSL/0.9.8q
Allow:
Content-Length: 353
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>405 Method Not Allowed</title>
```

OPTIONS to verify…

```
$ telnet www.cs.odu.edu 80
Trying 128.82.4.2...
Connected to xenon.cs.odu.edu.
Escape character is '^]'.
OPTIONS / HTTP/1.1
Testing-1: Hello, Sailor!
Testing-2: 69,105
Host: www.cs.odu.edu
Connection: close

HTTP/1.1 200 OK
Date: Mon, 16 Jan 2012 23:54:43 GMT
Server: Apache/2.2.17 (Unix) PHP/5.3.5 mod_ssl/2.2.17
OpenSSL/0.9.8q
Allow: GET,HEAD,POST,OPTIONS
Content-Length: 0
Connection: close
Content-Type: text/html

Connection closed by foreign host.
```

TRACE still not implemented

```
$ openssl s_client -connect www.cs.odu.edu:443
[ssl deletia]
TRACE / HTTP/1.1
Testing-1: Hello, Sailor!
Testing-2: 69,105
Host: www.cs.odu.edu.edu.edu.edu
Connection: close

HTTP/1.1 405 Not Allowed
Server: nginx
Date: Wed, 05 Sep 2018 15:30:14 GMT
Content-Type: text/html
Transfer-Encoding: chunked
Connection: close

a6
<html>
<head><title>405 Not Allowed</title></head>
<body bgcolor="white">
<center><h1>405 Not Allowed</h1></center>
<hr><center>nginx</center>
</body>
</html>

0

closed
```

```
$ openssl s_client -connect www.cs.odu.edu:443
[ssl deletia]
OPTIONS / HTTP/1.1
Host: www.cs.odu.edu
Connection: close

HTTP/1.1 405 Not Allowed
Server: nginx
Date: Wed, 05 Sep 2018 15:31:43 GMT
Content-Type: text/html
Transfer-Encoding: chunked
Connection: close

a6
<html>
<head><title>405 Not Allowed</title></head>
<body bgcolor="white">
<center><h1>405 Not Allowed</h1></center>
<hr><center>nginx</center>
</body>
</html>

0

closed
```

But now neither is
OPTIONS
(sigh)

Even worse – not RFC 2616/7230 compliant (ugh)

```
$ openssl s_client -connect
www.cs.odu.edu:443
[ssl deletia]
OPTIONS * HTTP/1.1
Host: www.cs.odu.edu
Connection: close

HTTP/1.1 400 Bad Request
Server: nginx
Date: Wed, 05 Sep 2018 15:32:46 GMT
Content-Type: text/html
Connection: close

<html>
<head><title>400 Bad Request</title></head>
<body bgcolor="white">
<center><h1>400 Bad Request</h1></center>
<hr><center>nginx</center>
</body>
</html>
```

Can no longer fold lines (boo)

3.2.4. Field Parsing

…

No whitespace is allowed between the header field-name and colon. In the past, differences in the handling of such whitespace have led to security vulnerabilities in request routing and response handling. A server MUST reject any received request message that contains whitespace between a header field-name and colon with a response code of 400 (Bad Request). A proxy MUST remove any such whitespace from a response message before forwarding the message downstream.

A field value might be preceded and/or followed by optional whitespace (OWS); a single SP preceding the field-value is preferred for consistent readability by humans. The field value does not include any leading or trailing whitespace: OWS occurring before the first non-whitespace octet of the field value or after the last non-whitespace octet of the field value ought to be excluded by parsers when extracting the field value from a header field.

Historically, HTTP header field values could be extended over multiple lines by preceding each extra line with at least one space or horizontal tab (obs-fold). This specification deprecates such line folding except within the message/http media type (Section 8.3.1). …

Prohibited in 2616, but not stated this explicitly

A server that receives an obs-fold in a request message that is not within a message/http container MUST either reject the message by sending a 400 (Bad Request), preferably with a representation explaining that obsolete line folding is unacceptable, or replace each received obs-fold with one or more SP octets prior to interpreting the field value or forwarding the message downstream.

Folding made for better readability

```
HTTP/1.1 200 OK
Date: Thu, 21 Jan 2010 00:06:51 GMT
Server: Apache-Coyote/1.1
Memento-Datetime: Tue, 20 Mar 2001 13:36:10 GMT
Link: <http://a.example.org/>; rel="original timegate",
 <http://a.example.org/?version=all&style=timemap>
   ; rel="timemap"; type="application/link-format"
   ; from="Tue, 15 Sep 2000 11:28:26 GMT"
   ; until="Wed, 20 Jan 2010 09:34:33 GMT"
Content-Length: 23364
Content-Type: text/html;charset=utf-8
Connection: close
```

Becomes:

```
HTTP/1.1 200 OK
Date: Thu, 21 Jan 2010 00:06:51 GMT
Server: Apache-Coyote/1.1
Memento-Datetime: Tue, 20 Mar 2001 13:36:10 GMT
Link: <http://a.example.org/>; rel="original timegate", <http://a.example.org/?version=all...
Content-Length: 23364
Content-Type: text/html;charset=utf-8
Connection: close
```

Common Log Format

```
127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0" 200 2326
```

```
68.10.168.183 - - [16/Jan/2012:18:18:30 -0400] "GET /~mln/ HTTP/1.1" 200 5336
68.10.168.183 - - [16/Jan/2012:18:18:32 -0400] "HEAD /~mln/ HTTP/1.1" 200 5336
```

Most web servers are configured to include two additional fields (value of “Referer” and “User-agent” headers) in their access logs (AKA Combined Log Format)

See:

- http://en.wikipedia.org/wiki/Common_Log_Format
- http://httpd.apache.org/docs/2.4/logs.html#common

MIME Types

- Multipurpose Internet Mail Extensions
  - RFCs 2045, 2046
  - http://www.iana.org/assignments/media-types/
  - used to populate http “Content-Type” response headers (and later, request headers!)
- Although not part of http, web servers generally have a configurable method of mapping file extensions to MIME types, e.g.:

.jpeg, .jpg	image/jpeg
.pdf	application/pdf
.ppt	application/vnd.ms-powerpoint
.htm, .html	text/html

MIME is easy, right?

What are these Content-types?

```
$ ls -1
fairlane.gif
fairlane.jpeg
fairlane.png
fairlane.txt
index.html.de
index.html.en
index.html.es
index.html.ja.jis
index.html.ko.euc-kr
index.html.ru.koi8-r
type-map.example
vt-uva.html.gz
vt-uva.html.Z
```

```
$ cat type-map.example
URI: index.html.de
Content-Language: de
Content-type: text/html; charset=ISO-8859-1

URI: index.html.en
Content-Language: en
Content-type: text/html; charset=ISO-8859-1

URI: index.html.es
Content-Language: es
Content-type: text/html; charset=ISO-8859-1

URI: index.html.ja.jis
Content-Language: ja
Content-type: text/html; charset=ISO-2022-JP

URI: index.html.ko.euc-kr
Content-Language: ko
Content-type: text/html; charset=EUC-KR

URI: index.html.ru.koi8-r
Content-Language: ru
Content-type: text/html; charset=KOI8-R
```

Don’t be like this…

```
$ curl -I https://www.cs.odu.edu/~mln/teaching/cs595-s12/a3-test/type-map.example
HTTP/1.1 200 OK
Server: nginx
Date: Wed, 05 Sep 2018 18:17:44 GMT
Content-Length: 520
Connection: keep-alive
Last-Modified: Sun, 12 Mar 2006 16:39:23 GMT
ETag: "208-40ecede54e4c0"
Accept-Ranges: bytes
Front-End-Https: on
```

This encourages “Content Sniffing”,
which is responsible for 37% of the evil in the world
https://en.wikipedia.org/wiki/Content_sniffing
and necessitated the rise of:

```
X-Content-Type-Options: nosniff
```

When in doubt, use:

```
Content-Type: application/octet-stream
```

Non-200 responses often have HTML message bodies, but you don’t always see them

```
$ curl -i http://bit.ly/2ogMITK
HTTP/1.1 301 Moved Permanently
Server: nginx
Date: Wed, 05 Sep 2018 18:28:47 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 161
Connection: keep-alive
Cache-Control: private, max-age=90
Location: http://ws-dl.blogspot.com/2016/10/2016-10-13-dodging-memory-hole-2016.html
Set-Cookie: _bit=i85isL-dac6ed8d52d00272b4-00t; Domain=bit.ly; Expires=Mon, 04 Mar 2019 18:28:47 GMT

<html>
<head><title>Bitly</title></head>
<body><a href="http://ws-dl.blogspot.com/2016/10/2016-10-13-dodging-memory-hole-2016.html">
moved here</a></body>
</html>
```

URIs are opaque

```
% curl -I https://en.wikipedia.org/wiki/File:DJ_Shadow_tim_festival.jpg
HTTP/1.1 200 OK
Date: Wed, 05 Sep 2018 18:44:55 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
Server: mw1248.eqiad.wmnet
X-Content-Type-Options: nosniff
…
Vary: Accept-Encoding,Cookie,Authorization
X-Powered-By: HHVM/3.18.6-dev
Content-language: en
Backend-Timing: D=146450 t=1536173095558659
Content-Encoding: gzip
X-Varnish: 50394305, 891111514
Via: 1.1 varnish (Varnish/5.1), 1.1 varnish (Varnish/5.1)
Accept-Ranges: bytes
Age: 0
X-Cache: cp1077 miss, cp1079 miss
X-Cache-Status: miss
…
Cache-Control: private, s-maxage=0, max-age=0, must-revalidate
Set-Cookie: GeoIP=US:VA:Norfolk:36.93:-76.24:v4; Path=/; secure; Domain=.wikipedia.org
```

Good things to have in a config file(s)…

- MIME types (including default type)
- document root
- your server name (version, assignment, etc.)
- location of your access log, debugging log(s)
- resolve IP address for logs --> hostnames (y/n)
- default port # your server will run on
- templates (or pointers to templates) for directory listings, dynamically created entities (e.g., for non-200 responses), log formats
- default timeout (seconds)
- virtual URIs
- redirects (code, URI1, URI2)