Web Server Design

Lecture 6 – Character, Content, and Transfer Encodings

Old Dominion University

Department of Computer Science

CS 431/531 Fall 2019

Sawood Alam <salam@cs.odu.edu>

2019-10-03

Original slides by Michael L. Nelson

HTTP equivalent of “they’re / their / there”, “you’re / your”, etc.

Extending the analogy, “ur” is acceptable only when you know the rules, but breaking them provides some measurable comfort or convenience.

http://theoatmeal.com/comics/misspelling

Encoding Can Mean Many Things

- Character encoding
  - “charset” attribute for textual MIME types
  - “utf-8” is the most popular charset, but there are many others
- Content encoding
- Transfer encoding

ASCII and Extended ASCII

- Character encoding is a mapping of a set of characters to a set of numbers
- American Standard Code for Information Interchange encodes various control and printable (lower/upper-case English letters, digits, and symbols) characters
- ASCII uses 7 bits (encodes 128 characters)
- In various Extended ASCII schemes remaining one bit (of a byte) is used to encode things like mathematical symbols

ASCII Table

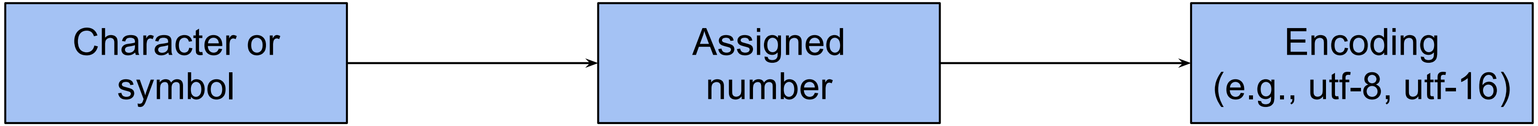
Binary	Hex	Dec	Char	Binary	Hex	Dec	Char	Binary	Hex	Dec	Char	Binary	Hex	Dec	Char
0000000	00	0	NUL (null)	0100000	20	32	SPACE	1000000	40	64	@	1100000	60	96	`
0000001	01	1	SOH (start of heading)	0100001	21	33	!	1000001	41	65	A	1100001	61	97	a
0000010	02	2	STX (start of text)	0100010	22	34	"	1000010	42	66	B	1100010	62	98	b
0000011	03	3	ETX (end of text)	0100011	23	35	#	1000011	43	67	C	1100011	63	99	c
0000100	04	4	EOT (end of transmission)	0100100	24	36	$	1000100	44	68	D	1100100	64	100	d
0000101	05	5	ENQ (enquiry)	0100101	25	37	%	1000101	45	69	E	1100101	65	101	e
0000110	06	6	ACK (acknowledge)	0100110	26	38	&	1000110	46	70	F	1100110	66	102	f
0000111	07	7	BEL (bell)	0100111	27	39	'	1000111	47	71	G	1100111	67	103	g
0001000	08	8	BS (backspace)	0101000	28	40	(	1001000	48	72	H	1101000	68	104	h
0001001	09	9	TAB (horizontal tab)	0101001	29	41	)	1001001	49	73	I	1101001	69	105	i
0001010	0A	10	LF (NL line feed, new line)	0101010	2A	42	*	1001010	4A	74	J	1101010	6A	106	j
0001011	0B	11	VT (vertical tab)	0101011	2B	43	+	1001011	4B	75	K	1101011	6B	107	k
0001100	0C	12	FF (NP form feed, new page)	0101100	2C	44	,	1001100	4C	76	L	1101100	6C	108	l
0001101	0D	13	CR (carriage return)	0101101	2D	45	-	1001101	4D	77	M	1101101	6D	109	m
0001110	0E	14	SO (shift out)	0101110	2E	46	.	1001110	4E	78	N	1101110	6E	110	n
0001111	0F	15	SI (shift in)	0101111	2F	47	/	1001111	4F	79	O	1101111	6F	111	o
0010000	10	16	DLE (data link escape)	0110000	30	48	0	1010000	50	80	P	1110000	70	112	p
0010001	11	17	DC1 (device control 1)	0110001	31	49	1	1010001	51	81	Q	1110001	71	113	q
0010010	12	18	DC2 (device control 2)	0110010	32	50	2	1010010	52	82	R	1110010	72	114	r
0010011	13	19	DC3 (device control 3)	0110011	33	51	3	1010011	53	83	S	1110011	73	115	s
0010100	14	20	DC4 (device control 4)	0110100	34	52	4	1010100	54	84	T	1110100	74	116	t
0010101	15	21	NAK (negative acknowledge)	0110101	35	53	5	1010101	55	85	U	1110101	75	117	u
0010110	16	22	SYN (synchronous idle)	0110110	36	54	6	1010110	56	86	V	1110110	76	118	v
0010111	17	23	ETB (end of trans. block)	0110111	37	55	7	1010111	57	87	W	1110111	77	119	w
0011000	18	24	CAN (cancel)	0111000	38	56	8	1011000	58	88	X	1111000	78	120	x
0011001	19	25	EM (end of medium)	0111001	39	57	9	1011001	59	89	Y	1111001	79	121	y
0011010	1A	26	SUB (substitute)	0111010	3A	58	:	1011010	5A	90	Z	1111010	7A	122	z
0011011	1B	27	ESC (escape)	0111011	3B	59	;	1011011	5B	91	[	1111011	7B	123	{
0011100	1C	28	FS (file separator)	0111100	3C	60	<	1011100	5C	92	\	1111100	7C	124	|
0011101	1D	29	GS (group separator)	0111101	3D	61	=	1011101	5D	93	]	1111101	7D	125	}
0011110	1E	30	RS (record separator)	0111110	3E	62	>	1011110	5E	94	^	1111110	7E	126	~
0011111	1F	31	US (unit separator)	0111111	3F	63	?	1011111	5F	95	_	1111111	7F	127	DEL

You Might Be Surprised to Know, There Exist Languages Other Than English

- Are 128 (or 256) symbols enough to represent every character in every language?
- What if every language comes with its own encoding (character to number mapping)?
  - Which they did, as a result we got hundreds of encodings
- Documents in one encoding become garbled in the other
  - This issue became more prominent on the Web
- How about multilingual documents?

Unicode to the Rescue

- Covers characters from 150+ modern and historic scripts
- Various symbol sets and emojis
- Supported by various modern platforms
- Evolving to encode more means of expressions
- Separates encoding scheme from numeric assignment



Character or symbol

Assigned number

Encoding (e.g., utf-8, utf-16)

UTF-16 and UTF-32

- UTF-32 is a fixed-width 4 byte encoding
  - Simple, but wasteful
- UTF-16 is a variable-length (16 or 32 bit) encoding
  - The two byte pairs of UTF-16 may appear in either order, depending on the implementation
    - This is called “endianness”
    - Denoted by Byte Order Mark (BOM) in the beginning
      - “0xFE 0xFF” for big-endian
      - “0xFF 0xFE” for little-endian

UTF-8

- Dynamic encoding
- ASCII encoding is a valid subset
- Currently uses 1 to 4 bytes, but can use up to 7 bytes

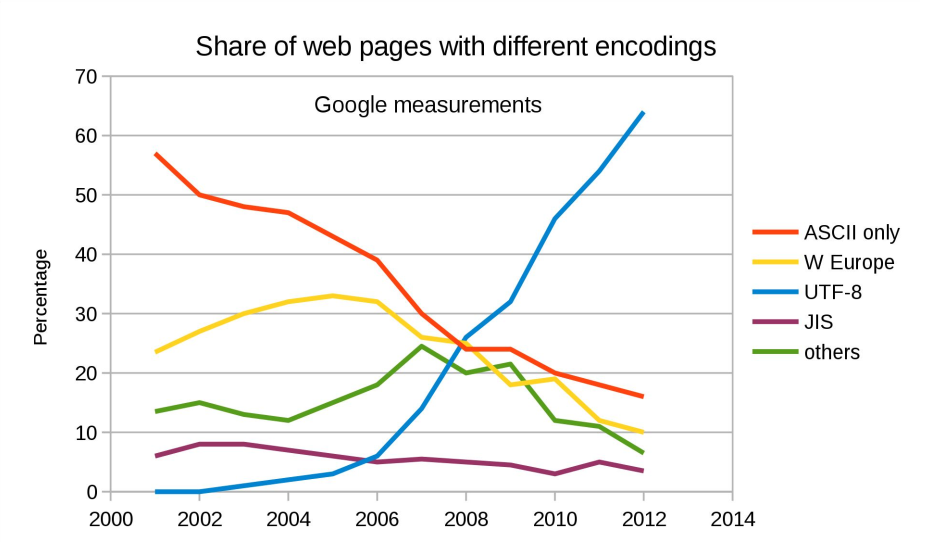
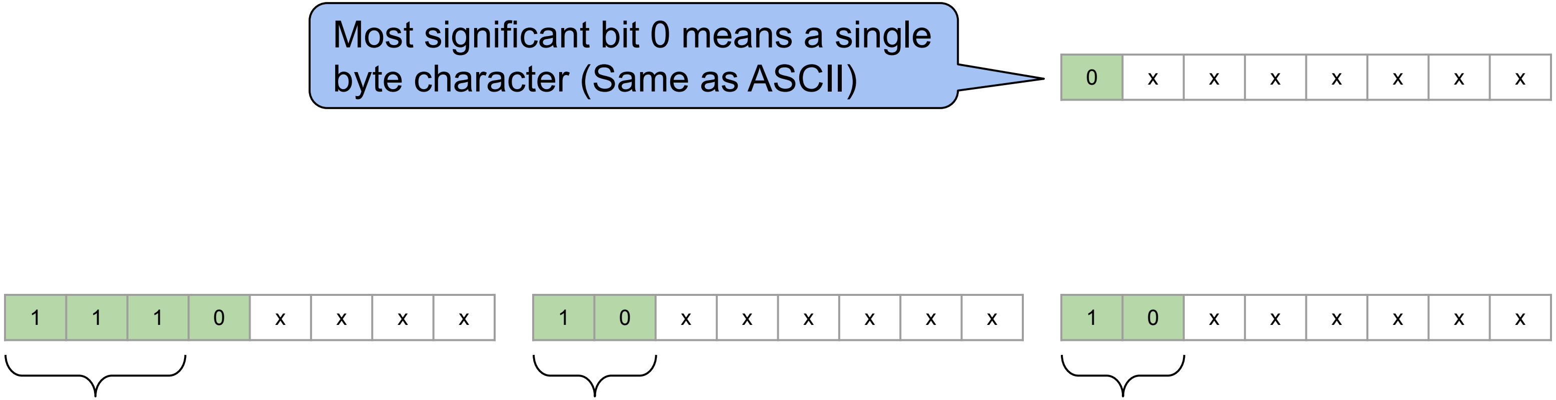


Image source: https://en.wikipedia.org/wiki/UTF-8

Now, the share of UTF-8 is above 94% on the Web
https://w3techs.com/technologies/details/en-utf8/all/all

UTF-8 Is the Most Elegant Encoding Hack



Most significant bit 0 means a single byte character (Same as ASCII)

0	x	x	x	x	x	x	x

1	1	1	0	x	x	x	x
1	0	x	x	x	x	x	x
1	0	x	x	x	x	x	x

Number of leading 1s mean the number of bytes for the character

Leading 10 does not mean a single byte, but a continuation mark

Common Content and Transfer Encodings

- identity
  - no encoding at all; defined in 2616, removed in 7230
- gzip
  - extension: .gz (sometimes seen as x-gzip, deprecated)
- compress
  - extension: .Z (sometimes seen as x-compress, deprecated)
- deflate
  - extension: .zip
- chunked
  - breaks the body into a series of server-chosen “chunks”
  - optimization for dynamically produced content

IANA registry, cf. “br” & “zstd”; http://www.iana.org/assignments/http-parameters/http-parameters.xhtml

Identity

- The default, “no transformation” encoding
  - even though it was removed in 7230 and never really existed in the wild, it is a useful rhetorical construct
  - “applying the identity encoding to a resource is an ____???____ operation”

    Hint: Applying identity encoding repeatedly makes no difference!

Content Codings

“Content coding values indicate an encoding transformation that has been or can be applied to a representation. Content codings are primarily used to allow a representation to be compressed or otherwise usefully transformed without losing the identity of its underlying media type and without loss of information. Frequently, the representation is stored in coded form, transmitted directly, and only decoded by the final recipient.”

- – 3.1.2.1, RFC 7231

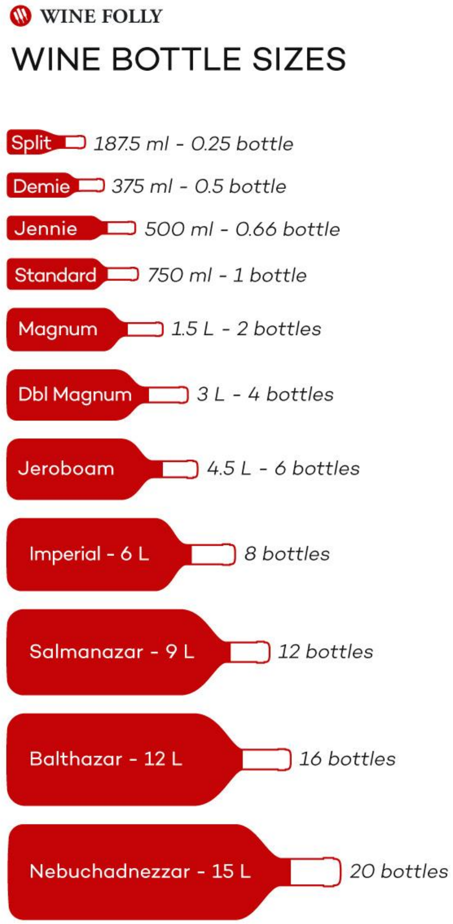
Content Encoding vs. Transfer Encoding

3.1.2.2, RFC 7231

Unlike Transfer-Encoding (Section 3.3.1 of [RFC7230]), the codings listed in Content-Encoding are a characteristic of the representation; the representation is defined in terms of the coded form, and all other metadata about the representation is about the coded form unless otherwise noted in the metadata definition. Typically, the representation is only decoded just prior to rendering or analogous usage.

If the media type includes an inherent encoding, such as a data format that is always compressed, then that encoding would not be restated in Content-Encoding even if it happens to be the same algorithm as one of the content codings.

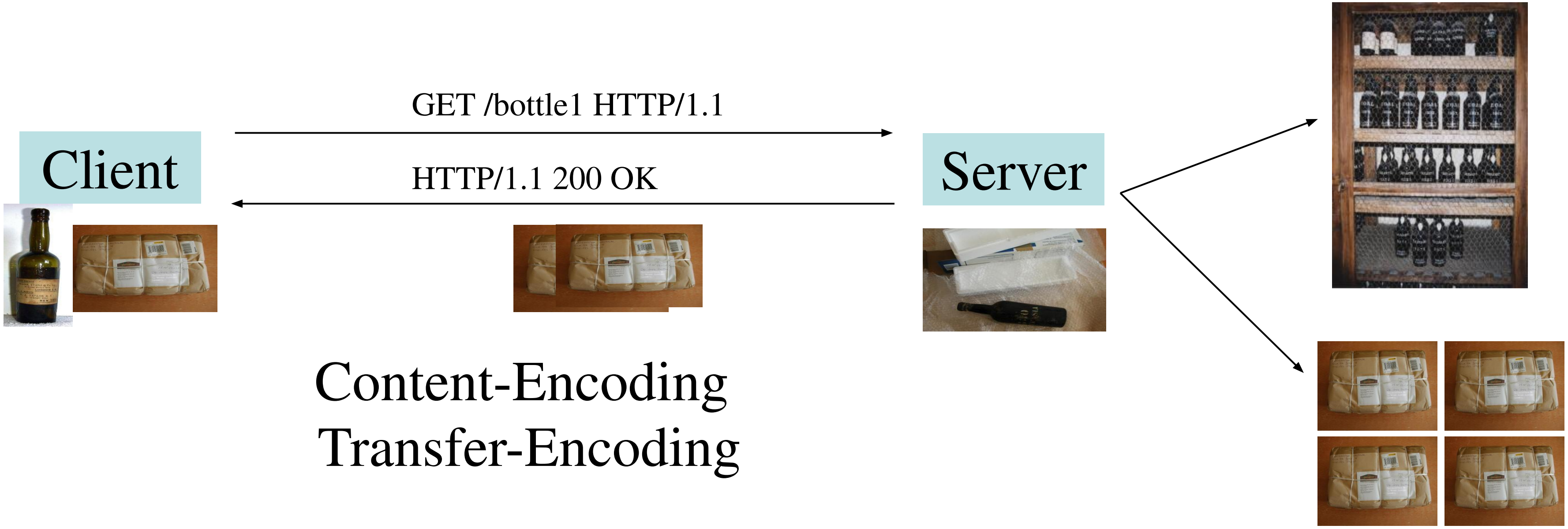
e.g., GIF uses LZW compression (“`compress`”),
but this is not reflected in a `Content-Encoding` header


WINE FOLLY
WINE BOTTLE SIZES
Split 187.5 ml - 0.25 bottle
Demie 375 ml - 0.5 bottle
Jennie 500 ml - 0.66 bottle
Standard 750 ml - 1 bottle
Magnum 1.5 L - 2 bottles
Dbl Magnum 3 L - 4 bottles
Jeroboam 4.5 L - 6 bottles
Imperial - 6 L 8 bottles
Salmanazar - 9 L 12 bottles
Balthazar - 12 L 16 bottles
Nebuchadnezzar - 15 L 20 bottles

https://winefolly.com/tutorial/wine-bottle-sizes/

The wine (liquid) is the `Content-type`;
the bottle size is the `Content-Encoding`

Content Encoding vs. Transfer Encoding



Client

GET /bottle1 HTTP/1.1

HTTP/1.1 200 OK

Server

Content-Encoding

Transfer-Encoding

Images from: http://www.madeirawineguide.com/madeira_v300/

Packaging/Shipping Doesn’t Change the Representation

https://mashable.com/2013/08/22/71-lbs-fedex/
https://about.van.fedex.com/blog/boeing-777-ecodemonstrator/

Content-Encoding Example (Correct)

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

HTTP/1.1 200 OK
Date: Mon, 20 Feb 2006 04:30:25 GMT
Server: Apache/1.3.26 (Unix) ApacheJServ/1.1.2
PHP/4.3.4
Last-Modified: Thu, 25 Jul 2002 16:58:58 GMT
ETag: "1c16-139ea-3d402e52"
Accept-Ranges: bytes
Content-Length: 80362
Connection: close
Content-Type: application/postscript
Content-Encoding: x-gzip

Connection closed by foreign host.
```

Content-Encoding Example (Incorrect)

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

HTTP/1.1 200 OK
Date: Mon, 26 Feb 2007 02:06:25 GMT
Server: Apache/2.2.0
Last-Modified: Thu, 25 Jul 2002 16:58:58 GMT
ETag: "1c16-139ea-92cab880"
Accept-Ranges: bytes
Content-Length: 80362
Connection: close
Content-Type: application/x-gzip

Connection closed by foreign host.
```

Wrong, Wrong, Wrong!!!!!!!

Why is it incorrect?

```
$ telnet www.cs.odu.edu 80
Trying 128.82.4.2...
Connected to xenon.cs.odu.edu.
Escape character is '^]'.
HEAD /~mln/adl98.ppt.gz HTTP/1.1
Host: www.cs.odu.edu
Connection: close

HTTP/1.1 200 OK
Date: Mon, 20 Feb 2012 02:21:29 GMT
Server: Apache/2.2.17 (Unix) PHP/5.3.5 mod_ssl/2.2.17
OpenSSL/0.9.8q
Last-Modified: Mon, 25 Mar 2002 17:15:44 GMT
ETag: "33e94-39d06961d5000"
Accept-Ranges: bytes
Content-Length: 212628
Connection: close
Content-Type: application/x-gzip
```

the encodings are the same, the types are different

Compress

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

HTTP/1.1 200 OK
Date: Mon, 20 Feb 2012 02:31:45 GMT
Server: Apache/2.2.17 (Unix) PHP/5.3.5 mod_ssl/2.2.17
OpenSSL/0.9.8q
Last-Modified: Thu, 12 Jun 2003 18:45:46 GMT
ETag: "7fffffff-3bfeb99a3ca80"
Accept-Ranges: bytes
Content-Length: 2147483647
Connection: close
Content-Type: application/x-compress
```

Zip

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

HTTP/1.1 200 OK
Date: Mon, 20 Feb 2012 03:16:35 GMT
Server: Apache/2.2.17 (Unix) PHP/5.3.5 mod_ssl/2.2.17
OpenSSL/0.9.8q
Last-Modified: Fri, 17 Feb 2012 20:43:13 GMT
ETag: "223e5a1-4b92efe4dfc40"
Accept-Ranges: bytes
Content-Length: 35906977
Connection: close
Content-Type: application/zip
```

Transfer Encodings

Unlike Content-Encoding (Section 3.1.2.1 of [RFC7231]), Transfer-Encoding is a property of the message, not of the representation, and any recipient along the request/response chain MAY decode the received transfer coding(s) or apply additional transfer coding(s) to the message body, assuming that corresponding changes are made to the Transfer-Encoding field-value.

3.3.1, RFC 7230

Chunked Encoding

“The chunked transfer coding wraps the payload body in order to transfer it as a series of chunks, each with its own size indicator, followed by an OPTIONAL trailer containing header fields. Chunked enables content streams of unknown size to be transferred as a sequence of length-delimited buffers, which enables the sender to retain connection persistence and the recipient to know when it has received the entire message.”

4.1, RFC 7230

Chunked Encoding Example

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

Connection closed by foreign host.
HTTP/1.1 301 Moved Permanently
Date: Mon, 09 Jan 2006 19:32:24 GMT
Server: Apache/1.3.26 (Unix) ApacheJServ/1.1.2 PHP/4.3.4
Location: http://www.cs.odu.edu/~mln/
Connection: close
Transfer-Encoding: chunked
Content-Type: text/html; charset=iso-8859-1

12e
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<HTML><HEAD>
<TITLE>301 Moved Permanently</TITLE>
</HEAD><BODY>
<H1>Moved Permanently</H1>
The document has moved <A HREF="http://www.cs.odu.edu/~mln/">here</A>.<P>
<HR>
<ADDRESS>Apache/1.3.26 Server at www.cs.odu.edu Port 80</ADDRESS>
</BODY></HTML>

0
```

Chunked Encoding Example 2

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

HTTP/1.1 200 OK
Date: Tue, 21 Feb 2006 03:54:31 GMT
Server: Apache/1.3.26 (Unix) ApacheJServ/1.1.2 PHP/4.3.4
Transfer-Encoding: chunked
Content-Type: text/html

5f6
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<!-- saved from url=(0036)http://www.cs.odu.edu/newcssite/new/ -->
<!-- saved from url=(0019)http://sci.odu.edu/ -->
<HTML xmlns:st1 = "urn:schemas-microsoft-com:office:smarttags">
<HEAD><TITLE>Department Of Computer Science</TITLE>

[lots of html deleted]
[demo this example to see the various  “chunks”]

0
```

Chunks visible in raw session

```
$ openssl s_client -connect www.cs.odu.edu:443
CONNECTED(00000003)
[much ssl deletia]
GET / HTTP/1.1
Host: www.cs.odu.edu
Connection: close

HTTP/1.1 200 OK
Server: nginx
Date: Wed, 17 Oct 2018 16:44:12 GMT
Content-Type: text/html
Transfer-Encoding: chunked
Connection: close

d3
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
<title>Redirect</title>
<meta http-equiv="REFRESH" content="0;url=http://www.cs.odu.edu"></HEAD>
<BODY>
Redirecting!
</BODY>
</HTML>


0

closed
```

User agents will sometimes hide the transfer-encoding chunk sizes!

```
$ curl -i www.cs.odu.edu
HTTP/1.1 200 OK
Server: nginx
Date: Wed, 17 Oct 2018 16:46:59 GMT
Content-Type: text/html
Transfer-Encoding: chunked
Connection: keep-alive

<html>
<meta http-equiv="refresh" content="0; URL='http://odu.edu/compsci'" />
</html>
```

Transfer-Encoding header is present, but curl has suppressed the byte sizes (still hidden with “-v” too)

```
$ curl -I www.cs.odu.edu
HTTP/1.1 200 OK
Server: nginx
Date: Wed, 17 Oct 2018 16:47:42 GMT
Content-Type: text/html
Connection: keep-alive
```

HEAD is less useful as well;
we don’t know the representation size (both Content-Length and Transfer-Encoding are absent)

Multiple Transfer encodings are possible, but chunked (if present) is always the last

3.3.1, RFC 7230

The Transfer-Encoding header field lists the transfer coding names corresponding to the sequence of transfer codings that have been (or will be) applied to the payload body in order to form the message body.
[…]
If any transfer coding other than chunked is applied to a request payload body, the sender MUST apply chunked as the final transfer coding to ensure that the message is properly framed. If any transfer coding other than chunked is applied to a response payload body, the sender MUST either apply chunked as the final transfer coding or terminate the message by closing the connection.

For example,

Transfer-Encoding: gzip, chunked indicates that the payload body has been compressed using the gzip coding and then chunked using the chunked coding while forming the message body.

TE Request Header & Transfer-Encoding Response Header

- Client specifies preferences for transfer encoding in the `TE` header
  - 4.3, RFC 7230
- Server marks the encoding used with the `Transfer-Encoding` header
  - 3.3.1, RFC 7230
- Both headers use the same encoding values available with `Content-Encoding`, plus the special `chunked` encoding and the `Trailers` value

Trailers requested, but server isn’t sending…

```
$ telnet www.cs.odu.edu 80
Trying 128.82.4.2...
Connected to xenon.cs.odu.edu.
Escape character is '^]'.
GET / HTTP/1.1
TE: gzip;q=1.0, Trailers
Host: www.cs.odu.edu

HTTP/1.1 200 OK
Date: Mon, 27 Feb 2006 15:52:33 GMT
Server: Apache/1.3.26 (Unix) ApacheJServ/1.1.2 PHP/4.3.4
Transfer-Encoding: chunked
Content-Type: text/html

5f6
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<!-- saved from url=(0036)http://www.cs.odu.edu/newcssite/new/ -->
<!-- saved from url=(0019)http://sci.odu.edu/ -->
[more html deleted]
```

Time / Space Tradeoff

- Hard to find examples of compression used in transfer encoding
  - https://web.archive.org/web/20001019025549/http://webreference.com:80/internet/software/servers/ https://web.archive.org/web/20001019025549/http://webreference.com:80/internet/software/servers/ https://web.archive.org/web/20001019025549/http://webreference.com:80/internet/software/servers/ (read w/ curl or lynx; js rendering fails) https://web.archive.org/web/20050114061329/http://www-128.ibm.com/developerworks/web/library/wa-httpcomp/
  - idea: for very heavy volume web servers, answering the request quickly is more important than preserving bandwidth
- Complexity of management seems to be the limiting factor in compression with content encodings

Trailer Response Header

- The “Trailer” response header lets the client know that additional headers will appear at the end of the chunked response
  - sections 4.1.2, 4.4, RFC 7230
  - headers can be reconstructed by downstream servers
  - headers that can never be trailers:
    - Transfer-Encoding
    - Content-Length
    - Trailer

Trailer Example

```
HTTP/1.1 200 OK
Date: Mon, 22 Mar 2004 11:15:03 GMT
Content-Type: text/html
Content-Length: 129
Expires: Sat, 27 Mar 2004 21:12:00 GMT

<html><body><p>The file you requested is
3,400 bytes long
and was last modified: Sat, 20 Mar 2004
21:12:00 GMT.
</p></body></html>
```

```
HTTP/1.1 200 OK
Date: Mon, 22 Mar 2004 11:15:03 GMT
Content-Type: text/html
Transfer-Encoding: chunked
Trailer: Expires

29
<html><body><p>The file you requested is
5
3,400
23
bytes long and was last modified:
1d
Sat, 20 Mar 2004 21:12:00 GMT
13
.</p></body></html>
0
Expires: Sat, 27 Mar 2004 21:12:00 GMT
```

“Expires:” response header covered in section 5.3, RFC 7234

Examples from: http://www.tcpipguide.com/free/t_HTTPDataLengthIssuesChunkedTransfersandMessageTrai-3.htm

Two More Request Headers to Process

Referer Request Header

5.5.2, RFC 7231

The "Referer" [sic] header field allows the user agent to specify a URI reference for the resource from which the target URI was obtained (i.e., the "referrer", though the field name is misspelled). A user agent MUST NOT include the fragment and userinfo components of the URI reference [RFC3986], if any, when generating the Referer field value.

Example:

```
Referer: http://www.example.org/hypertext/Overview.html
```

cf. `rel=“noreferrer”` (also note the correct spelling)
https://stackoverflow.com/questions/50773152/when-should-i-use-rel-noreferrer

User-Agent Request Header

5.5.3, RFC 7231

The "User-Agent" header field contains information about the user agent originating the request, which is often used by servers to help identify the scope of reported interoperability problems, to work around or tailor responses to avoid particular user agent limitations, and for analytics regarding browser or operating system use. A user agent SHOULD send a User-Agent field in each request unless specifically configured not to do so.

Example:

```
User-Agent: CERN-LineMode/2.15 libwww/2.17b3
```

Lying with User-Agent

```
$ curl -I "https://www.amazon.com/Mountain-Has-Fallen-EP/dp/B073JS3Y9Q/"
HTTP/1.1 503 Service Unavailable
Content-Type: text/html
Content-Length: 2671
Connection: keep-alive
Server: Server
Date: Wed, 17 Oct 2018 17:19:09 GMT
[deletia]

$ curl -I -A "mozilla"  "https://www.amazon.com/Mountain-Has-Fallen-EP/dp/B073JS3Y9Q/"
HTTP/1.1 405 Method Not Allowed
Content-Type: text/html;charset=UTF-8
Connection: keep-alive
Server: Server
Date: Wed, 17 Oct 2018 17:19:32 GMT
[deletia]

$ curl -i -A "mozilla"  -s "https://www.amazon.com/Mountain-Has-Fallen-EP/dp/B073JS3Y9Q/" |  head -10
HTTP/1.1 200 OK
Content-Type: text/html;charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
Server: Server
Date: Wed, 17 Oct 2018 17:20:21 GMT
Strict-Transport-Security: max-age=47474747; includeSubDomains; preload
Vary: Accept-Encoding,User-Agent,X-Amazon-CDN-Cache
P3P: policyref="https://www.amazon.com/w3c/p3p.xml",CP="CAO DSP LAW CUR ADM IVAo IVDo CONo OTPo OUR
DELi PUBi OTRi BUS PHY ONL UNI PUR FIN COM NAV INT DEM CNT STA HEA PRE LOC GOV OTC "
Cache-Control: no-cache, no-transform
```

I’m an iPhone, I swear!

```
$ curl -ILs https://en.wikipedia.org/ | grep -iE "^(http|location)"
HTTP/2 301
location: https://en.wikipedia.org/wiki/Main_Page
HTTP/2 200

$ curl -ILs -A "iphone" https://en.wikipedia.org/ | grep -iE "^(http|location)"
HTTP/2 301
location: https://en.wikipedia.org/wiki/Main_Page
HTTP/2 302
location: https://en.m.wikipedia.org/wiki/Main_Page
HTTP/2 200
```

cf. https://www.business2community.com/instagram/post-instagram-computer-02013790