URI = scheme ":" hier-part ["?" query] ["#" fragment]
hier-part = "//" authority path-abempty / path-absolute / path-rootless / path-empty
URI-reference = URI / relative-ref
absolute-URI = scheme ":" hier-part ["?" query]
relative-ref = relative-part ["?" query] ["#" fragment]
relative-part = "//" authority path-abempty / path-absolute / path-noscheme / path-empty
scheme = ALPHA * (ALPHA / DIGIT / "+" / "-" / ".")
authority = [userinfo "@"] host [":" port]
userinfo = * (unreserved / pct-encoded / sub-delims / ":")
host = IP-literal / IPv4address / reg-name
port = * DIGIT
IP-literal = "[" (IPv6address / IPvFuture) "]"
IPvFuture = "v" 1* HEXDIG "." 1* (unreserved / sub-delims / ":")
IPv6address = 6 (h16 ":") ls32 / "::" 5 (h16 ":") ls32 / [h16] "::" 4 (h16 ":") ls32 / [h16 *1 (":" h16)] "::" 3 (h16 ":") ls32 / [h16 *2 (":" h16)] "::" 2 (h16 ":") ls32 / [h16 *3 (":" h16)] "::" h16 ":" ls32 / [h16 *4 (":" h16)] "::" ls32 / [h16 *5 (":" h16)] "::" h16 / [h16 *6 (":" h16)] "::"
h16 = 1*4 HEXDIG
ls32 = (h16 ":" h16) / IPv4address
IPv4address = dec-octet "." dec-octet "." dec-octet "." dec-octet
dec-octet = "25" %x30-35 / "2" %x30-34 DIGIT / "1" 2 DIGIT / %x31-39 DIGIT / DIGIT
reg-name = * (unreserved / pct-encoded / sub-delims)
path = path-abempty / path-absolute / path-noscheme / path-rootless / path-empty
path-abempty = * ("/" segment)
path-absolute = "/" [segment-nz * ("/" segment)]
path-noscheme = segment-nz-nc * ("/" segment)
path-rootless = segment-nz * ("/" segment)
path-empty = ""
segment = * pchar
segment-nz = 1* pchar
segment-nz-nc = 1* (unreserved / pct-encoded / sub-delims / "@")
pchar = unreserved / pct-encoded / sub-delims / ":" / "@"
query = * (pchar / "/" / "?")
fragment = * (pchar / "/" / "?")
pct-encoded = "%" HEXDIG HEXDIG
unreserved = ALPHA / DIGIT / "-" / "." / "_" / "~"
reserved = gen-delims / sub-delims
gen-delims = ":" / "/" / "?" / "#" / "[" / "]" / "@"
sub-delims = "!" / "$" / "&" / "'" / "(" / ")" / "*" / "+" / "," / ";" / "="
language-range = (1*8 ALPHA * ("-" 1*8 alphanum)) / "*"
alphanum = ALPHA / DIGIT
Language-Tag = langtag / privateuse / grandfathered
langtag = language ["-" script] ["-" region] * ("-" variant) * ("-" extension) ["-" privateuse]

language = 2*3 ALPHA ["-" extlang] / 4 ALPHA / 5*8 ALPHA
extlang = 3 ALPHA *2 ("-" 3 ALPHA)
script = 4 ALPHA
region = 2 ALPHA / 3 DIGIT
variant = 5*8 alphanum / (DIGIT 3 alphanum)
extension = singleton 1* ("-" (2*8 alphanum))
singleton = DIGIT / %x41-57 / %x59-5A / %x61-77 / %x79-7A
privateuse = "x" 1* ("-" (1*8 alphanum))
grandfathered = irregular / regular
irregular = "en-GB-oed" / "i-ami" / "i-bnn" / "i-default" / "i-enochian" / "i-hak" / "i-klingon" / "i-lux" / "i-mingo" / "i-navajo" / "i-pwn" / "i-tao" / "i-tay" / "i-tsu" / "sgn-BE-FR" / "sgn-BE-NL" / "sgn-CH-DE"
regular = "art-lojban" / "cel-gaulish" / "no-bok" / "no-nyn" / "zh-guoyu" / "zh-hakka" / "zh-min" / "zh-min-nan" / "zh-xiang"
BWS = OWS
Connection = * ("," OWS) connection-option * (OWS "," [OWS connection-option])
Content-Length = 1* DIGIT
HTTP-message = start-line * (header-field CRLF) CRLF [message-body]
HTTP-name = %x48.54.54.50
HTTP-version = HTTP-name "/" DIGIT "." DIGIT
Host = uri-host [":" port]
OWS = * (SP / HTAB)
RWS = 1* (SP / HTAB)
TE = [("," / t-codings) * (OWS "," [OWS t-codings])]
Trailer = * ("," OWS) field-name * (OWS "," [OWS field-name])
Transfer-Encoding = * ("," OWS) transfer-coding * (OWS "," [OWS transfer-coding])
Upgrade = * ("," OWS) protocol * (OWS "," [OWS protocol])
Via = * ("," OWS) (received-protocol RWS received-by [RWS comment]) * (OWS "," [OWS (received-protocol RWS received-by [RWS comment])])
absolute-form = absolute-URI
absolute-path = 1* ("/" segment)
asterisk-form = "*"
authority-form = authority
chunk = chunk-size [chunk-ext] CRLF chunk-data CRLF
chunk-data = 1* OCTET
chunk-ext = * (";" chunk-ext-name ["=" chunk-ext-val])
chunk-ext-name = token
chunk-ext-val = token / quoted-string
chunk-size = 1* HEXDIG
chunked-body = * chunk last-chunk trailer-part CRLF
comment = "(" * (ctext / quoted-pair / comment) ")"
connection-option = token
ctext = HTAB / SP / %x21-27 / %x2A-5B / %x5D-7E / obs-text
field-content = field-vchar [1* (SP / HTAB) field-vchar]

field-name = token
field-value = * (field-content / obs-fold)
field-vchar = VCHAR / obs-text
http-URI = "http://" authority path-abempty ["?" query] ["#" fragment]
https-URI = "https://" authority path-abempty ["?" query] ["#" fragment]
last-chunk = 1* "0" [chunk-ext] CRLF
message-body = * OCTET
method = token
obs-fold = CRLF 1* (SP / HTAB)
obs-text = %x80-FF
origin-form = absolute-path ["?" query]
partial-URI = relative-part ["?" query]
protocol = protocol-name ["/" protocol-version]
protocol-name = token
protocol-version = token
pseudonym = token
qdtext = HTAB / SP / "!" / %x23-5B / %x5D-7E / obs-text
quoted-pair = "\" (HTAB / SP / VCHAR / obs-text)
quoted-string = DQUOTE * (qdtext / quoted-pair) DQUOTE
rank = ("0" ["." *3 DIGIT]) / ("1" ["." *3 "0"])
reason-phrase = * (HTAB / SP / VCHAR / obs-text)
received-by = (uri-host [":" port]) / pseudonym
received-protocol = [protocol-name "/"] protocol-version
request-line = method SP request-target SP HTTP-version CRLF
request-target = origin-form
start-line = request-line / status-line
status-code = 3 DIGIT
status-line = HTTP-version SP status-code SP reason-phrase CRLF
t-codings = "trailers" / (transfer-coding [t-ranking])
t-ranking = OWS ";" OWS "q=" rank
tchar = "!" / "#" / "$" / "%" / "&" / "'" / "*" / "+" / "-" / "." / "^" / "_" / "`" / "|" / "~" / DIGIT / ALPHA
token = 1* tchar
trailer-part = * (header-field CRLF)
transfer-coding = "chunked" / "compress" / "deflate" / "gzip" / transfer-extension
transfer-extension = token * (OWS ";" OWS transfer-parameter)
transfer-parameter = token BWS "=" BWS (token / quoted-string)
uri-host = host
Accept = [("," / (media-range [accept-params])) * (OWS "," [OWS (media-range [accept-params])])]
Accept-Charset = * ("," OWS) ((charset / "*") [weight]) * (OWS "," [OWS ((charset / "*") [weight])])
Accept-Encoding = [("," / (codings [weight])) * (OWS "," [OWS (codings [weight])])]
Accept-Language = * ("," OWS) (language-range [weight]) * (OWS "," [OWS (language-range [weight])])

Allow = [("," / method) * (OWS "," [OWS method])]
Content-Encoding = * ("," OWS) content-coding * (OWS "," [OWS content-coding])
Content-Language = * ("," OWS) Language-Tag * (OWS "," [OWS Language-Tag])
Content-Location = absolute-URI / partial-URI
Content-Type = media-type
Date = HTTP-date
Expect = "100-continue"
GMT = %x47.4D.54
HTTP-date = IMF-fixdate / obs-date
IMF-fixdate = day-name "," SP date1 SP time-of-day SP GMT
Location = URI-reference
Max-Forwards = 1* DIGIT
Referer = absolute-URI / partial-URI
Retry-After = HTTP-date / delay-seconds
Server = product * (RWS (product / comment))
User-Agent = product * (RWS (product / comment))
Vary = "*" / (* ("," OWS) field-name * (OWS "," [OWS field-name]))
accept-ext = OWS ";" OWS token ["=" (token / quoted-string)]
accept-params = weight * accept-ext
asctime-date = day-name SP date3 SP time-of-day SP year
charset = token
codings = content-coding / "identity" / "*"
content-coding = token
date1 = day SP month SP year
date2 = day "-" month "-" 2 DIGIT
date3 = month SP (2 DIGIT / (SP DIGIT))
day = 2 DIGIT
day-name = %x4D.6F.6E / %x54.75.65 / %x57.65.64 / %x54.68.75 / %x46.72.69 / %x53.61.74 / %x53.75.6E
day-name-l = %x4D.6F.6E.64.61.79 / %x54.75.65.73.64.61.79 / %x57.65.64.6E.65.73.64.61.79 / %x54.68.75.72.73.64.61.79 / %x46.72.69.64.61.79 / %x53.61.74.75.72.64.61.79 / %x53.75.6E.64.61.79
delay-seconds = 1* DIGIT
hour = 2 DIGIT
media-range = ("*/*" / (type "/" subtype) / (type "/*")) * (OWS ";" OWS parameter)
media-type = type "/" subtype * (OWS ";" OWS parameter)
minute = 2 DIGIT
month = %x4A.61.6E / %x46.65.62 / %x4D.61.72 / %x41.70.72 / %x4D.61.79 / %x4A.75.6E / %x4A.75.6C / %x41.75.67 / %x53.65.70 / %x4F.63.74 / %x4E.6F.76 / %x44.65.63
obs-date = rfc850-date / asctime-date
parameter = token "=" (token / quoted-string)
product = token ["/" product-version]
product-version = token
qvalue = ("0" ["." *3 DIGIT]) / ("1" ["." *3 "0"])

rfc850-date = day-name-l "," SP date2 SP time-of-day SP GMT
second = 2 DIGIT
subtype = token
time-of-day = hour ":" minute ":" second
type = token
weight = OWS ";" OWS "q=" qvalue
year = 4 DIGIT
ETag = entity-tag
If-Match = "*" / (* ("," OWS) entity-tag * (OWS "," [OWS entity-tag]))
If-Modified-Since = HTTP-date
If-None-Match = "*" / (* ("," OWS) entity-tag * (OWS "," [OWS entity-tag]))
If-Unmodified-Since = HTTP-date
Last-Modified = HTTP-date
entity-tag = [weak] opaque-tag
etagc = "!" / %x23-7E
opaque-tag = DQUOTE * etagc DQUOTE
weak = %x57.2F
Accept-Ranges = acceptable-ranges
Content-Range = byte-content-range / other-content-range
If-Range = entity-tag / HTTP-date
Range = byte-ranges-specifier / other-ranges-specifier
acceptable-ranges = (* ("," OWS) range-unit * (OWS "," [OWS range-unit])) / "none"
byte-content-range = bytes-unit SP (byte-range-resp / unsatisfied-range)
byte-range = first-byte-pos "-" last-byte-pos
byte-range-resp = byte-range "/" (complete-length / "*")
byte-range-set = * ("," OWS) (byte-range-spec / suffix-byte-range-spec) * (OWS "," [OWS (byte-range-spec / suffix-byte-range-spec)])
byte-range-spec = first-byte-pos "-" [last-byte-pos]
byte-ranges-specifier = bytes-unit "=" byte-range-set
bytes-unit = "bytes"
complete-length = 1* DIGIT
first-byte-pos = 1* DIGIT
last-byte-pos = 1* DIGIT
other-content-range = other-range-unit SP other-range-resp
other-range-resp = * CHAR
other-range-set = 1* VCHAR
other-range-unit = token
other-ranges-specifier = other-range-unit "=" other-range-set
range-unit = bytes-unit / other-range-unit
suffix-byte-range-spec = "-" suffix-length
suffix-length = 1* DIGIT
unsatisfied-range = "*/" complete-length

Age = delta-seconds
Cache-Control = * ("," OWS) cache-directive * (OWS "," [OWS cache-directive])
Expires = HTTP-date
Pragma = * ("," OWS) pragma-directive * (OWS "," [OWS pragma-directive])
Warning = * ("," OWS) warning-value * (OWS "," [OWS warning-value])
cache-directive = token ["=" (token / quoted-string)]
delta-seconds = 1* DIGIT
extension-pragma = token ["=" (token / quoted-string)]
pragma-directive = "no-cache" / extension-pragma
warn-agent = (uri-host [":" port]) / pseudonym
warn-code = 3 DIGIT
warn-date = DQUOTE HTTP-date DQUOTE
warn-text = quoted-string
warning-value = warn-code SP warn-agent SP warn-text [SP warn-date]
Proxy-Authenticate = * ("," OWS) challenge * (OWS "," [OWS challenge])
Proxy-Authorization = credentials
WWW-Authenticate = * ("," OWS) challenge * (OWS "," [OWS challenge])
auth-param = token BWS "=" BWS (token / quoted-string)
auth-scheme = token
challenge = auth-scheme [1* SP (token68 / [("," / auth-param) * (OWS "," [OWS auth-param])])]
credentials = auth-scheme [1* SP (token68 / [("," / auth-param) * (OWS "," [OWS auth-param])])]
Authorization = credentials
token68 = 1* (ALPHA / DIGIT / "-" / "." / "_" / "~" / "+" / "/") * "="
Connection-header = "Connection" ":" OWS Connection OWS
Content-Length-header = "Content-Length" ":" OWS Content-Length OWS
Content-Type-header = "Content-Type" ":" OWS Content-Type OWS
Trailer-header = "Trailer" ":" OWS Trailer OWS
Transfer-Encoding-header = "Transfer-Encoding" ":" OWS Transfer-Encoding OWS
Upgrade-header = "Upgrade" ":" OWS Upgrade OWS
Via-header = "Via" ":" OWS Via OWS
Age-header = "Age" ":" OWS Age OWS
Expires-header = "Expires" ":" OWS Expires OWS
Date-header = "Date" ":" OWS Date OWS
Location-header = "Location" ":" OWS Location OWS
Retry-After-header = "Retry-After" ":" OWS Retry-After OWS
Vary-header = "Vary" ":" OWS Vary OWS
Warning-header = "Warning" ":" OWS Warning OWS
Cache-Control-header = "Cache-Control" ":" OWS Cache-Control OWS
Expect-header = "Expect" ":" OWS Expect OWS
Host-header = "Host" ":" OWS Host OWS
Max-Forwards-header = "Max-Forwards" ":" OWS Max-Forwards OWS

Pragma-header = "Pragma" ":" OWS Pragma OWS
Range-header = "Range" ":" OWS Range OWS
TE-header = "TE" ":" OWS TE OWS
If-Match-header = "If-Match" ":" OWS If-Match OWS
If-None-Match-header = "If-None-Match" ":" OWS If-None-Match OWS
If-Modified-Since-header = "If-Modified-Since" ":" OWS If-Modified-Since OWS
If-Unmodified-Since-header = "If-Unmodified-Since" ":" OWS If-Unmodified-Since OWS
If-Range-header = "If-Range" ":" OWS If-Range OWS
Accept-header = "Accept" ":" OWS Accept OWS
Accept-Charset-header = "Accept-Charset" ":" OWS Accept-Charset OWS
Accept-Encoding-header = "Accept-Encoding" ":" OWS Accept-Encoding OWS
Accept-Language-header = "Accept-Language" ":" OWS Accept-Language OWS
Authorization-header = "Authorization" ":" OWS Authorization OWS
Proxy-Authorization-header = "Proxy-Authorization" ":" OWS Proxy-Authorization OWS
Referer-header = "Referer" ":" OWS Referer OWS
User-Agent-header = "User-Agent" ":" OWS User-Agent OWS
cookie-pair = cookie-name "=" cookie-value
cookie-name = token
cookie-value = (DQUOTE * cookie-octet DQUOTE) / * cookie-octet
cookie-octet = %x21 / %x23-2B / %x2D-3A / %x3C-5B / %x5D-7E
Cookie-header = "Cookie:" OWS cookie-string OWS
cookie-string = cookie-pair * (";" SP cookie-pair)
header-field = Connection-header / Content-Length-header / Content-Type-header / Cookie-header / Transfer-Encoding-header / Expect-header / Host-header / Accept-header / Accept-Charset-header / Accept-Encoding-header / Accept-Language-header / Referer-header / User-Agent-header / (field-name ":" OWS field-value OWS)