HTTPbis Working Group	R. Fielding, Ed.
Internet-Draft	Day Software
Obsoletes: 2616 (if approved)	J. Gettys
Intended status: Standards Track	Alcatel-Lucent
Expires: February 5, 2011	J. Mogul
	HP
	H. Frystyk
	Microsoft
	L. Masinter
	Adobe Systems
	P. Leach
	Microsoft
	T. Berners-Lee
	W3C/MIT
	Y. Lafon, Ed.
	W3C
	J. Reschke, Ed.
	greenbytes
	August 4, 2010

TOC

HTTP/1.1, part 5: Range Requests and Partial Responses
draft-ietf-httpbis-p5-range-11

Abstract

The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems. HTTP has been in use by the World Wide Web global information initiative since 1990. This document is Part 5 of the seven-part specification that defines the protocol referred to as "HTTP/1.1" and, taken together, obsoletes RFC 2616. Part 5 defines range-specific requests and the rules for constructing and combining responses to those requests.

Editorial Note (To be removed by RFC Editor)

Discussion of this draft should take place on the HTTPBIS working group mailing list (ietf-http-wg@w3.org). The current issues list is at http://tools.ietf.org/wg/httpbis/trac/report/3 and related documents (including fancy diffs) can be found at http://tools.ietf.org/wg/httpbis/.

The changes in this draft are summarized in Appendix D.12 (Since draft-ietf-httpbis-p5-range-10).

Status of This Memo

This Internet-Draft will expire on February 5, 2011.

Copyright Notice

Table of Contents

TOC

1. Introduction

HTTP clients often encounter interrupted data transfers as a result of cancelled requests or dropped connections. When a cache has stored a partial representation, it is desirable to request the remainder of that representation in a subsequent request rather than transfer the entire representation. There are also a number of Web applications that benefit from being able to request only a subset of a larger representation, such as a single page of a very large document or only part of an image to be rendered by a device with limited local storage.

This document defines HTTP/1.1 range requests, partial responses, and the multipart/byteranges media type. The protocol for range requests is an OPTIONAL feature of HTTP, designed so resources or recipients that do not implement this feature can respond as if it is a normal GET request without impacting interoperability. Partial responses are indicated by a distinct status code to not be mistaken for full responses by intermediate caches that might not implement the feature.

Although the HTTP range request mechanism is designed to allow for extensible range types, this specification only defines requests for byte ranges.

1.1. Requirements

TOC

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] (Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” March 1997.).

An implementation is not compliant if it fails to satisfy one or more of the "MUST" or "REQUIRED" level requirements for the protocols it implements. An implementation that satisfies all the "MUST" or "REQUIRED" level and all the "SHOULD" level requirements for its protocols is said to be "unconditionally compliant"; one that satisfies all the "MUST" level requirements but not all the "SHOULD" level requirements for its protocols is said to be "conditionally compliant".

1.2. Syntax Notation

TOC

This specification uses the ABNF syntax defined in Section 1.2 of [Part1] (Fielding, R., Ed., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., Berners-Lee, T., Lafon, Y., Ed., and J. Reschke, Ed., “HTTP/1.1, part 1: URIs, Connections, and Message Parsing,” August 2010.) (which extends the syntax defined in [RFC5234] (Crocker, D., Ed. and P. Overell, “Augmented BNF for Syntax

Specifications: ABNF," January 2008.) with a list rule). Appendix C (Collected ABNF) shows the collected ABNF, with the list rule expanded. The following core rules are included by reference, as defined in [RFC5234] (Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF," January 2008.), Appendix B.1: ALPHA (letters), CR (carriage return), CRLF (CR LF), CTL (controls), DIGIT (decimal 0-9), DQUOTE (double quote), HEXDIG (hexadecimal 0-9/A-F/a-f), LF (line feed), OCTET (any 8-bit sequence of data), SP (space), VCHAR (any visible USASCII character), and WSP (whitespace).

1.2.1. Core Rules

TOC

The core rules below are defined in Section 1.2.2 of [Part1] (Fielding, R., Ed., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., Berners-Lee, T., Lafon, Y., Ed., and J. Reschke, Ed., "HTTP/1.1, part 1: URIs, Connections, and Message Parsing," August 2010.):

```
  token      = <token, defined in [Part1], Section 1.2.2>
  OWS        = <OWS, defined in [Part1], Section 1.2.2>
```

1.2.2. ABNF Rules defined in other Parts of the Specification

TOC

The ABNF rules below are defined in other parts:

```
  HTTP-date  = <HTTP-date, defined in [Part1], Section 6.1>


  entity-tag = <entity-tag, defined in [Part4], Section 2>
```

2. Range Units

TOC

HTTP/1.1 allows a client to request that only part (a range of) the representation be included within the response. HTTP/1.1 uses range units in the Range (Section 5.4 (Range)) and Content-Range (Section 5.2 (Content-Range)) header fields. A representation can be broken down into subranges according to various structural units.

```
range-unit       = bytes-unit / other-range-unit
bytes-unit       = "bytes"
other-range-unit = token
```

HTTP/1.1 has been designed to allow implementations of applications that do not depend on knowledge of ranges. The only range unit defined by HTTP/1.1 is "bytes". Additional specifiers can be defined as described in Section 2.1 (Range Specifier Registry).
If a range unit is not understood in a request, a server MUST ignore the whole Range header (Section 5.4 (Range)). If a range unit is not understood in a response, an intermediary SHOULD pass the response to the client; a client MUST fail.

2.1. Range Specifier Registry

TOC

The HTTP Ranger Specifier Registry defines the name space for the range specifier names.
Registrations MUST include the following fields:

- Name
- Description
- Pointer to specification text

Values to be added to this name space are subject to IETF review ([RFC5226] (Narten, T. and H. Alvestrand, “Guidelines for Writing an IANA Considerations Section in RFCs,” May 2008.), Section 4.1).
The registry itself is maintained at http://www.iana.org/assignments/http-range-specifiers.

3. Status Code Definitions

TOC

3.1. 206 Partial Content

TOC

The server has fulfilled the partial GET request for the resource. The request MUST have included a Range header field (Section 5.4 (Range)) indicating the desired range, and MAY have included an If-Range header field (Section 5.3 (If-Range)) to make the request conditional.

The response MUST include the following header fields:

- *Either a Content-Range header field (Section 5.2 (Content-Range)) indicating the range included with this response, or a multipart/byteranges Content-Type including Content-Range fields for each part. If a Content-Length header field is present in the response, its value MUST match the actual number of octets transmitted in the message-body.

- *Date

- *Cache-Control, ETag, Expires, Content-Location, Last-Modified, and/or Vary, if the header field would have been sent in a 200 response to the same request

If the 206 response is the result of an If-Range request, the response SHOULD NOT include other representation header fields. Otherwise, the response MUST include all of the representation header fields that would have been returned with a 200 (OK) response to the same request.
A cache MUST NOT combine a 206 response with other previously cached content if the ETag or Last-Modified headers do not match exactly, see Section 4 (Combining Ranges).
A cache that does not support the Range and Content-Range headers MUST NOT cache 206 (Partial Content) responses. Furthermore, if a response uses a range unit that is not understood by the cache, then it MUST NOT be cached either.

3.2. 416 Requested Range Not Satisfiable

TOC

A server SHOULD return a response with this status code if a request included a Range request-header field (Section 5.4 (Range)), and none of the ranges-specifier values in this field overlap the current extent of the selected resource, and the request did not include an If-Range request-header field (Section 5.3 (If-Range)). (For byte-ranges, this means that the first-byte-pos of all of the byte-range-spec values were greater than the current length of the selected resource.)
When this status code is returned for a byte-range request, the response SHOULD include a Content-Range header field specifying the current length of the representation (see Section 5.2 (Content-Range)). This response MUST NOT use the multipart/byteranges content-type.

TOC

4. Combining Ranges

A response might transfer only a subrange of a representation, either because the request included one or more Range specifications, or because a connection closed prematurely. After several such transfers, a cache might have received several ranges of the same representation. If a cache has a stored non-empty set of subranges for a representation, and an incoming response transfers another subrange, the cache MAY combine the new subrange with the existing set if both the following conditions are met:

- Both the incoming response and the cache entry have a cache validator.

- The two cache validators match using the strong comparison function (see Section 4 of [Part4] (Fielding, R., Ed., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., Berners-Lee, T., Lafon, Y., Ed., and J. Reschke, Ed., “HTTP/1.1, part 4: Conditional Requests,” August 2010.)).

If either requirement is not met, the cache MUST use only the most recent partial response (based on the Date values transmitted with every response, and using the incoming response if these values are equal or missing), and MUST discard the other partial information.

5. Header Field Definitions

TOC

This section defines the syntax and semantics of HTTP/1.1 header fields related to range requests and partial responses.

5.1. Accept-Ranges

TOC

The "Accept-Ranges" response-header field allows a resource to indicate its acceptance of range requests.

```
Accept-Ranges     = "Accept-Ranges" ":" OWS Accept-Ranges-v
Accept-Ranges-v   = acceptable-ranges
acceptable-ranges = 1#range-unit / "none"
```

Origin servers that accept byte-range requests MAY send

```
Accept-Ranges: bytes
```

but are not required to do so. Clients MAY generate range requests without having received this header for the resource involved. Range units are defined in Section 2 (Range Units).
Servers that do not accept any kind of range request for a resource MAY send

```
       Accept-Ranges: none
```

to advise the client not to attempt a range request.

5.2. Content-Range

TOC

The "Content-Range" header field is sent with a partial representation to specify where in the full representation the payload body is intended to be applied.
Range units are defined in Section 2 (Range Units).

```
       Content-Range = "Content-Range" ":" OWS Content-Range-v
       Content-Range-v = content-range-spec

       content-range-spec      = byte-content-range-spec
                               / other-content-range-spec
       byte-content-range-spec = bytes-unit SP
                                 byte-range-resp-spec "/"
                                 ( instance-length / "*" )

       byte-range-resp-spec    = (first-byte-pos "-" last-byte-pos)
                               / "*"

       instance-length         = 1*DIGIT

       other-content-range-spec = other-range-unit SP
                                  other-range-resp-spec
       other-range-resp-spec    = *CHAR
```

The header SHOULD indicate the total length of the full representation, unless this length is unknown or difficult to determine. The asterisk "*" character means that the instance-length is unknown at the time when the response was generated.
Unlike byte-ranges-specifier values (see Section 5.4.1 (Byte Ranges)), a byte-range-resp-spec MUST only specify one range, and MUST contain absolute byte positions for both the first and last byte of the range.
A byte-content-range-spec with a byte-range-resp-spec whose last-byte-pos value is less than its first-byte-pos value, or whose instance-length value is less than or equal to its last-byte-pos value, is invalid. The recipient of an invalid byte-content-range-spec MUST ignore it and any content transferred along with it.

In the case of a byte range request: A server sending a response with status code 416 (Requested range not satisfiable) SHOULD include a Content-Range field with a byte-range-resp-spec of "*". The instance-length specifies the current length of the selected resource. A response with status code 206 (Partial Content) MUST NOT include a Content-Range field with a byte-range-resp-spec of "*".
Examples of byte-content-range-spec values, assuming that the representation contains a total of 1234 bytes:

- The first 500 bytes:

  ```
  bytes 0-499/1234
  ```

- The second 500 bytes:

  ```
  bytes 500-999/1234
  ```

- All except for the first 500 bytes:

  ```
  bytes 500-1233/1234
  ```

- The last 500 bytes:

  ```
  bytes 734-1233/1234
  ```

When an HTTP message includes the content of a single range (for example, a response to a request for a single range, or to a request for a set of ranges that overlap without any holes), this content is transmitted with a Content-Range header, and a Content-Length header showing the number of bytes actually transferred. For example,

```
HTTP/1.1 206 Partial Content
Date: Wed, 15 Nov 1995 06:25:24 GMT
Last-Modified: Wed, 15 Nov 1995 04:58:08 GMT
Content-Range: bytes 21010-47021/47022
Content-Length: 26012
Content-Type: image/gif
```

When an HTTP message includes the content of multiple ranges (for example, a response to a request for multiple non-overlapping ranges), these are transmitted as a multipart message. The multipart media type used for this purpose is "multipart/byteranges" as defined in Appendix A (Internet Media Type multipart/byteranges).
A response to a request for a single range MUST NOT be sent using the multipart/byteranges media type. A response to a request for multiple ranges, whose result is a single range, MAY be sent as a multipart/

byteranges media type with one part. A client that cannot decode a multipart/byteranges message MUST NOT ask for multiple ranges in a single request.

When a client requests multiple ranges in one request, the server SHOULD return them in the order that they appeared in the request.

If the server ignores a byte-range-spec because it is syntactically invalid, the server SHOULD treat the request as if the invalid Range header field did not exist. (Normally, this means return a 200 response containing the full representation).

If the server receives a request (other than one including an If-Range request-header field) with an unsatisfiable Range request-header field (that is, all of whose byte-range-spec values have a first-byte-pos value greater than the current length of the selected resource), it SHOULD return a response code of 416 (Requested range not satisfiable) (Section 3.2 (416 Requested Range Not Satisfiable)).

> Note: Clients cannot depend on servers to send a 416 (Requested range not satisfiable) response instead of a 200 (OK) response for an unsatisfiable Range request-header, since not all servers implement this request-header.

5.3. If-Range

TOC

If a client has a partial copy of a representation in its cache, and wishes to have an up-to-date copy of the entire representation in its cache, it could use the Range request-header with a conditional GET (using either or both of If-Unmodified-Since and If-Match.) However, if the condition fails because the representation has been modified, the client would then have to make a second request to obtain the entire current representation.

The "If-Range" request-header field allows a client to "short-circuit" the second request. Informally, its meaning is "if the representation is unchanged, send me the part(s) that I am missing; otherwise, send me the entire new representation".

```
  If-Range   = "If-Range" ":" OWS If-Range-v
  If-Range-v = entity-tag / HTTP-date
```

If the client has no entity-tag for a representation, but does have a Last-Modified date, it MAY use that date in an If-Range header. (The server can distinguish between a valid HTTP-date and any form of entity-tag by examining no more than two characters.) The If-Range header SHOULD only be used together with a Range header, and MUST be ignored if the request does not include a Range header, or if the server does not support the sub-range operation.

If the entity-tag given in the If-Range header matches the current cache validator for the representation, then the server SHOULD provide the specified sub-range of the representation using a 206 (Partial Content) response. If the cache validator does not match, then the server SHOULD return the entire representation using a 200 (OK) response.

5.4. Range

TOC

5.4.1. Byte Ranges

TOC

Since all HTTP representations are transferred as sequences of bytes, the concept of a byte range is meaningful for any HTTP representation. (However, not all clients and servers need to support byte-range operations.)

Byte range specifications in HTTP apply to the sequence of bytes in the representation body (not necessarily the same as the message-body).

A byte range operation MAY specify a single range of bytes, or a set of ranges within a single representation.

```
       byte-ranges-specifier = bytes-unit "=" byte-range-set
       byte-range-set  = 1#( byte-range-spec / suffix-byte-range-spec )
       byte-range-spec = first-byte-pos "-" [ last-byte-pos ]
       first-byte-pos  = 1*DIGIT
       last-byte-pos   = 1*DIGIT
```

The first-byte-pos value in a byte-range-spec gives the byte-offset of the first byte in a range. The last-byte-pos value gives the byte-offset of the last byte in the range; that is, the byte positions specified are inclusive. Byte offsets start at zero.

If the last-byte-pos value is present, it MUST be greater than or equal to the first-byte-pos in that byte-range-spec, or the byte-range-spec is syntactically invalid. The recipient of a byte-range-set that includes one or more syntactically invalid byte-range-spec values MUST ignore the header field that includes that byte-range-set.

If the last-byte-pos value is absent, or if the value is greater than or equal to the current length of the representation body, last-byte-pos is taken to be equal to one less than the current length of the representation in bytes.

By its choice of last-byte-pos, a client can limit the number of bytes retrieved without knowing the size of the representation.

```
suffix-byte-range-spec = "-" suffix-length
suffix-length = 1*DIGIT
```

A suffix-byte-range-spec is used to specify the suffix of the representation body, of a length given by the suffix-length value. (That is, this form specifies the last N bytes of a representation.) If the representation is shorter than the specified suffix-length, the entire representation is used.

If a syntactically valid byte-range-set includes at least one byte-range-spec whose first-byte-pos is less than the current length of the representation, or at least one suffix-byte-range-spec with a non-zero suffix-length, then the byte-range-set is satisfiable. Otherwise, the byte-range-set is unsatisfiable. If the byte-range-set is unsatisfiable, the server SHOULD return a response with a 416 (Requested range not satisfiable) status code. Otherwise, the server SHOULD return a response with a 206 (Partial Content) status code containing the satisfiable ranges of the representation.

Examples of byte-ranges-specifier values (assuming a representation of length 10000):

*The first 500 bytes (byte offsets 0-499, inclusive):

```
bytes=0-499
```

*The second 500 bytes (byte offsets 500-999, inclusive):

```
bytes=500-999
```

*The final 500 bytes (byte offsets 9500-9999, inclusive):

```
bytes=-500
```

Or:

```
bytes=9500-
```

*The first and last bytes only (bytes 0 and 9999):

```
bytes=0-0,-1
```

*Several legal but not canonical specifications of the second 500 bytes (byte offsets 500-999, inclusive):

```
                bytes=500-600,601-999
                bytes=500-700,601-999
```

5.4.2. Range Retrieval Requests

TOC

The "Range" request-header field defines the GET method (conditional or not) to request one or more sub-ranges of the response representation body, instead of the entire representation body.

```
  Range   = "Range" ":" OWS Range-v
  Range-v = byte-ranges-specifier
          / other-ranges-specifier
  other-ranges-specifier = other-range-unit "=" other-range-set
  other-range-set = 1*CHAR
```

A server MAY ignore the Range header. However, HTTP/1.1 origin servers and intermediate caches ought to support byte ranges when possible, since Range supports efficient recovery from partially failed transfers, and supports efficient partial retrieval of large representations.
If the server supports the Range header and the specified range or ranges are appropriate for the representation:

- The presence of a Range header in an unconditional GET modifies what is returned if the GET is otherwise successful. In other words, the response carries a status code of 206 (Partial Content) instead of 200 (OK).

- The presence of a Range header in a conditional GET (a request using one or both of If-Modified-Since and If-None-Match, or one or both of If-Unmodified-Since and If-Match) modifies what is returned if the GET is otherwise successful and the condition is true. It does not affect the 304 (Not Modified) response returned if the conditional is false.

In some cases, it might be more appropriate to use the If-Range header (see Section 5.3 (If-Range)) in addition to the Range header.
If a proxy that supports ranges receives a Range request, forwards the request to an inbound server, and receives an entire representation in reply, it SHOULD only return the requested range to its client. It SHOULD store the entire received response in its cache if that is consistent with its cache allocation policies.

6. IANA Considerations

TOC

6.1. Status Code Registration

TOC

The HTTP Status Code Registry located at http://www.iana.org/assignments/http-status-codes shall be updated with the registrations below:

Value	Description	Reference
206	Partial Content	Section 3.1 (206 Partial Content)
416	Requested Range Not Satisfiable	Section 3.2 (416 Requested Range Not Satisfiable)

6.2. Header Field Registration

TOC

The Message Header Field Registry located at http://www.iana.org/assignments/message-headers/message-header-index.html shall be updated with the permanent registrations below (see [RFC3864] (Klyne, G., Nottingham, M., and J. Mogul, “Registration Procedures for Message Header Fields,” September 2004.)):

Header Field Name	Protocol	Status	Reference
Accept-Ranges	http	standard	Section 5.1 (Accept-Ranges)
Content-Range	http	standard	Section 5.2 (Content-Range)
If-Range	http	standard	Section 5.3 (If-Range)
Range	http	standard	Section 5.4 (Range)

The change controller is: "IETF (iesg@ietf.org) - Internet Engineering Task Force".

6.3. Range Specifier Registration

TOC

The registration procedure for HTTP Range Specifiers is defined by Section 2.1 (Range Specifier Registry) of this document.
The HTTP Range Specifier Registry shall be created at http://www.iana.org/assignments/http-range-specifiers and be populated with the registrations below:

Range Specifier Name	Description	Reference
bytes	a range of octets	(this specification)

The change controller is: "IETF (iesg@ietf.org) - Internet Engineering Task Force".

7. Security Considerations

TOC

No additional security considerations have been identified beyond those applicable to HTTP in general [Part1] (Fielding, R., Ed., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., Berners-Lee, T., Lafon, Y., Ed., and J. Reschke, Ed., “HTTP/1.1, part 1: URIs, Connections, and Message Parsing,” August 2010.).

8. Acknowledgments

TOC

Most of the specification of ranges is based on work originally done by Ari Luotonen and John Franks, with additional input from Steve Zilles, Daniel W. Connolly, Roy T. Fielding, Jim Gettys, Martin Hamilton, Koen Holtman, Shel Kaplan, Paul Leach, Alex Lopez-Ortiz, Larry Masinter, Jeff Mogul, Lou Montulli, David W. Morris, Luigi Rizzo, and Bill Weihl.

9. References

TOC

9.1. Normative References

TOC

[Part1]	Fielding, R., Ed., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., Berners-Lee, T., Lafon, Y., Ed., and J. Reschke, Ed., "HTTP/1.1, part 1: URIs, Connections, and Message Parsing," draft-ietf-httpbis-p1-messaging-11 (work in progress), August 2010.
[Part4]	Fielding, R., Ed., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., Berners-Lee, T., Lafon, Y., Ed., and J. Reschke, Ed., "HTTP/1.1, part 4: Conditional Requests," draft-ietf-httpbis-p4-conditional-11 (work in progress), August 2010.
[RFC2046]	Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types," RFC 2046, November 1996.
[RFC2119]	Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," BCP 14, RFC 2119, March 1997.
[RFC5234]	Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF," STD 68, RFC 5234, January 2008.

9.2. Informative References

TOC

[RFC2616]	Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1," RFC 2616, June 1999.
[RFC3864]	Klyne, G., Nottingham, M., and J. Mogul, "Registration Procedures for Message Header Fields," BCP 90, RFC 3864, September 2004.
[RFC4288]	Freed, N. and J. Klensin, "Media Type Specifications and Registration Procedures," BCP 13, RFC 4288, December 2005.
[RFC5226]	Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs," BCP 26, RFC 5226, May 2008.

Appendix A. Internet Media Type multipart/byteranges

TOC

When an HTTP 206 (Partial Content) response message includes the content of multiple ranges (a response to a request for multiple non-overlapping ranges), these are transmitted as a multipart message-body ([RFC2046] (Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types," November 1996.), Section

5.1). The media type for this purpose is called "multipart/byteranges". The following is to be registered with IANA [RFC4288] (Freed, N. and J. Klensin, “Media Type Specifications and Registration Procedures,” December 2005.).

> Note: Despite the name "multipart/byteranges" is not limited to the byte ranges only.

The multipart/byteranges media type includes one or more parts, each with its own Content-Type and Content-Range fields. The required boundary parameter specifies the boundary string used to separate each body-part.

> **Type name:** multipart
>
> **Subtype name:** byteranges
>
> **Required parameters:** boundary
>
> **Optional parameters:** none
>
> **Encoding considerations:** only "7bit", "8bit", or "binary" are permitted
>
> **Security considerations:** none
>
> **Interoperability considerations:** none
>
> **Published specification:** This specification (see Appendix A (Internet Media Type multipart/byteranges)).
>
> **Applications that use this media type:**
> **Additional information:**
>
> > **Magic number(s):** none
> >
> > **File extension(s):** none
> >
> > **Macintosh file type code(s):** none
>
> **Person and email address to contact for further information:** See Authors Section.
>
> **Intended usage:** COMMON
>
> **Restrictions on usage:** none
>
> **Author/Change controller:** IESG

For example:

```
HTTP/1.1 206 Partial Content
Date: Wed, 15 Nov 1995 06:25:24 GMT
Last-Modified: Wed, 15 Nov 1995 04:58:08 GMT
Content-type: multipart/byteranges; boundary=THIS_STRING_SEPARATES

--THIS_STRING_SEPARATES
Content-type: application/pdf
Content-range: bytes 500-999/8000

...the first range...
--THIS_STRING_SEPARATES
Content-type: application/pdf
Content-range: bytes 7000-7999/8000

...the second range
--THIS_STRING_SEPARATES--
```

Other example:

```
HTTP/1.1 206 Partial Content
Date: Tue, 14 Nov 1995 06:25:24 GMT
Last-Modified: Tue, 14 July 04:58:08 GMT
Content-type: multipart/byteranges; boundary=THIS_STRING_SEPARATES

--THIS_STRING_SEPARATES
Content-type: video/example
Content-range: exampleunit 1.2-4.3/25

...the first range...
--THIS_STRING_SEPARATES
Content-type: video/example
Content-range: exampleunit 11.2-14.3/25

...the second range
--THIS_STRING_SEPARATES--
```

Notes:

1. Additional CRLFs MAY precede the first boundary string in the body.

2. Although [RFC2046] (Freed, N. and N. Borenstein, “Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types,” November 1996.) permits the boundary string to be quoted, some existing implementations handle a quoted boundary string incorrectly.

3. A number of browsers and servers were coded to an early draft of the byteranges specification to use a media type of

multipart/x-byteranges, which is almost, but not quite compatible with the version documented in HTTP/1.1.

Appendix B. Compatibility with Previous Versions

TOC

B.1. Changes from RFC 2616

TOC

Clarify that it is not ok to use a weak cache validator in a 206 response. (Section 3.1 (206 Partial Content))

Clarify that multipart/byteranges can consist of a single part. (Appendix A (Internet Media Type multipart/byteranges))

TOC

Appendix C. Collected ABNF

```
Accept-Ranges = "Accept-Ranges:" OWS Accept-Ranges-v
Accept-Ranges-v = acceptable-ranges

Content-Range = "Content-Range:" OWS Content-Range-v
Content-Range-v = content-range-spec

HTTP-date = <HTTP-date, defined in [Part1], Section 6.1>

If-Range = "If-Range:" OWS If-Range-v
If-Range-v = entity-tag / HTTP-date

OWS = <OWS, defined in [Part1], Section 1.2.2>

Range = "Range:" OWS Range-v
Range-v = byte-ranges-specifier / other-ranges-specifier

acceptable-ranges = ( *( "," OWS ) range-unit *( OWS "," [ OWS
 range-unit ] ) ) / "none"

byte-content-range-spec = bytes-unit SP byte-range-resp-spec "/" (
 instance-length / "*" )
byte-range-resp-spec = ( first-byte-pos "-" last-byte-pos ) / "*"
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"

content-range-spec = byte-content-range-spec /
 other-content-range-spec

entity-tag = <entity-tag, defined in [Part4], Section 2>

first-byte-pos = 1*DIGIT

instance-length = 1*DIGIT

last-byte-pos = 1*DIGIT

other-content-range-spec = other-range-unit SP other-range-resp-spec
other-range-resp-spec = *CHAR
other-range-set = 1*CHAR
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

    token = <token, defined in [Part1], Section 1.2.2>
```

ABNF diagnostics:

```
    ; Accept-Ranges defined but not used
    ; Content-Range defined but not used
    ; If-Range defined but not used
    ; Range defined but not used
```

Appendix D. Change Log (to be removed by RFC Editor before publication)

TOC

D.1. Since RFC2616

TOC

Extracted relevant partitions from [RFC2616] (Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, “Hypertext Transfer Protocol -- HTTP/1.1,” June 1999.).

D.2. Since draft-ietf-httpbis-p5-range-00

TOC

Closed issues:

- http://tools.ietf.org/wg/httpbis/trac/ticket/18: "Cache validators in 206 responses" (http://purl.org/NET/http-errata#ifrange206)

- http://tools.ietf.org/wg/httpbis/trac/ticket/35: "Normative and Informative references"

- http://tools.ietf.org/wg/httpbis/trac/ticket/86: "Normative up-to-date references"

TOC

D.3.   Since draft-ietf-httpbis-p5-range-01

Closed issues:

- http://tools.ietf.org/wg/httpbis/trac/ticket/55: "Updating to RFC4288"

Ongoing work on ABNF conversion (http://tools.ietf.org/wg/httpbis/trac/ticket/36):

- Add explicit references to BNF syntax and rules imported from other parts of the specification.

D.4.   Since draft-ietf-httpbis-p5-range-02

TOC

Ongoing work on IANA Message Header Registration (http://tools.ietf.org/wg/httpbis/trac/ticket/40):

- Reference RFC 3984, and update header registrations for headers defined in this document.

D.5.   Since draft-ietf-httpbis-p5-range-03

TOC

D.6.   Since draft-ietf-httpbis-p5-range-04

TOC

Closed issues:

- http://tools.ietf.org/wg/httpbis/trac/ticket/133: "multipart/byteranges minimum number of parts"

Ongoing work on ABNF conversion (http://tools.ietf.org/wg/httpbis/trac/ticket/36):

- Use "/" instead of "|" for alternatives.

- Introduce new ABNF rules for "bad" whitespace ("BWS"), optional whitespace ("OWS") and required whitespace ("RWS").

- Rewrite ABNFs to spell out whitespace rules, factor out header value format definitions.

D.7. Since draft-ietf-httpbis-p5-range-05

TOC

Closed issues:

- http://tools.ietf.org/wg/httpbis/trac/ticket/142: "State base for *-byte-pos and suffix-length"

Ongoing work on Custom Ranges (http://tools.ietf.org/wg/httpbis/trac/ticket/85):

- Remove bias in favor of byte ranges; allow custom ranges in ABNF.

Final work on ABNF conversion (http://tools.ietf.org/wg/httpbis/trac/ticket/36):

- Add appendix containing collected and expanded ABNF, reorganize ABNF introduction.

D.8. Since draft-ietf-httpbis-p5-range-06

TOC

Closed issues:

- http://tools.ietf.org/wg/httpbis/trac/ticket/161: "base for numeric protocol elements"

D.9. Since draft-ietf-httpbis-p5-range-07

TOC

Closed issues:

- Fixed discrepancy in the If-Range definition about allowed validators.
- http://tools.ietf.org/wg/httpbis/trac/ticket/150: "multipart/byteranges for custom range units"

- http://tools.ietf.org/wg/httpbis/trac/ticket/151: "range unit missing from other-ranges-specifier in Range header"
- http://tools.ietf.org/wg/httpbis/trac/ticket/198: "move IANA registrations for optional status codes"

D.10. Since draft-ietf-httpbis-p5-range-08

TOC

No significant changes.

D.11. Since draft-ietf-httpbis-p5-range-09

TOC

No significant changes.

D.12. Since draft-ietf-httpbis-p5-range-10

TOC

Closed issues:

- http://tools.ietf.org/wg/httpbis/trac/ticket/69: "Clarify 'Requested Variant'"
- http://tools.ietf.org/wg/httpbis/trac/ticket/109: "Clarify entity / representation / variant terminology"
- http://tools.ietf.org/wg/httpbis/trac/ticket/220: "consider removing the 'changes from 2068' sections"

Ongoing work on Custom Ranges (http://tools.ietf.org/wg/httpbis/trac/ticket/85):

- Add IANA registry.

Index

TOC

Authors' Addresses

TOC

	Roy T. Fielding (editor)
	Day Software
	23 Corporate Plaza DR, Suite 280
	Newport Beach, CA 92660
	USA
Phone:	+1-949-706-5300
Fax:	+1-949-706-5305
EMail:	fielding@gbiv.com
URI:	http://roy.gbiv.com/
	Jim Gettys
	Alcatel-Lucent Bell Labs
	21 Oak Knoll Road
	Carlisle, MA 01741
	USA
EMail:	jg@freedesktop.org
URI:	http://gettys.wordpress.com/
	Jeffrey C. Mogul
	Hewlett-Packard Company
	HP Labs, Large Scale Systems Group
	1501 Page Mill Road, MS 1177
	Palo Alto, CA 94304
	USA
EMail:	JeffMogul@acm.org
	Henrik Frystyk Nielsen
	Microsoft Corporation
	1 Microsoft Way
	Redmond, WA 98052
	USA
EMail:	henrikn@microsoft.com
	Larry Masinter
	Adobe Systems, Incorporated
	345 Park Ave
	San Jose, CA 95110
	USA
EMail:	LMM@acm.org

URI: http://larry.masinter.net/

Paul J. Leach
Microsoft Corporation
1 Microsoft Way
Redmond, WA 98052
EMail: paulle@microsoft.com

Tim Berners-Lee
World Wide Web Consortium
MIT Computer Science and Artificial Intelligence Laboratory
The Stata Center, Building 32
32 Vassar Street
Cambridge, MA 02139
USA
EMail: timbl@w3.org
URI: http://www.w3.org/People/Berners-Lee/

Yves Lafon (editor)
World Wide Web Consortium
W3C / ERCIM
2004, rte des Lucioles
Sophia-Antipolis, AM 06902
France
EMail: ylafon@w3.org
URI: http://www.raubacapeu.net/people/yves/

Julian F. Reschke (editor)
greenbytes GmbH
Hafenweg 16
Muenster, NW 48155
Germany
Phone: +49 251 2807760
Fax: +49 251 2807761
EMail: julian.reschke@greenbytes.de
URI: http://greenbytes.de/tech/webdav/