Network Working Group	R. Fielding, Ed.
Internet-Draft	Day Software
Obsoletes: 2616 (if approved)	J. Gettys
Intended status: Standards Track	One Laptop per Child
Expires: July 16, 2008	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
	January 13, 2008

TOC

HTTP/1.1, part 5: Range Requests and Partial Responses
draft-ietf-httpbis-p5-range-01

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.
Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.
Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as “work in progress.”
The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.
This Internet-Draft will expire on July 16, 2008.

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://www.tools.ietf.org/wg/httpbis/trac/report/11 and related documents (including fancy diffs) can be found at http://www.tools.ietf.org/wg/httpbis/.
This draft incorporates those issue resolutions that were either collected in the original RFC2616 errata list (http://purl.org/NET/http-errata), or which were agreed upon on the mailing list between October 2006 and November 2007 (as published in "draft-lafon-rfc2616bis-03").

Table of Contents

1. Introduction
    1.1. Requirements
2. Range Units
3. Status Code Definitions
    3.1. 206 Partial Content
    3.2. 416 Requested Range Not Satisfiable
4. Combining Byte Ranges
5. Header Field Definitions
    5.1. Accept-Ranges
    5.2. Content-Range
    5.3. If-Range
    5.4. Range
        5.4.1. Byte Ranges
        5.4.2. Range Retrieval Requests
6. IANA Considerations
7. Security Considerations
8. Acknowledgments
9. References

9.1. Normative References
9.2. Informative References
Appendix A. Internet Media Type multipart/byteranges
Appendix B. Compatibility with Previous Versions
B.1. Changes from RFC 2068
B.2. Changes from RFC 2616
Appendix C. Change Log (to be removed by RFC Editor before publication)
C.1. Since RFC2616
C.2. Since draft-ietf-httpbis-p5-range-00
§ Index
§ Authors' Addresses
§ Intellectual Property and Copyright Statements

---

1. Introduction

TOC

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."

---

2. Range Units

TOC

HTTP/1.1 allows a client to request that only part (a range of) the response entity 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. An entity 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
```

The only range unit defined by HTTP/1.1 is "bytes". HTTP/1.1 implementations MAY ignore ranges specified using other units. HTTP/1.1 has been designed to allow implementations of applications that do not depend on knowledge of ranges.

---

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

*ETag and/or Content-Location, if the header would have been sent in a 200 response to the same request

*Expires, Cache-Control, and/or Vary, if the field-value might differ from that sent in any previous response for the same variant

If the 206 response is the result of an If-Range request, the response SHOULD NOT include other entity-headers. Otherwise, the response MUST include all of the entity-headers 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 Byte Ranges).
A cache that does not support the Range and Content-Range headers MUST NOT cache 206 (Partial Content) responses.

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 range-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. (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 entity-header field specifying the current length of the selected resource (see Section 5.2 (Content-Range)). This response MUST NOT use the multipart/byteranges content-type.

---

4. Combining Byte Ranges

TOC

A response might transfer only a subrange of the bytes of an entity-body, either because the request included one or more Range specifications, or because a connection was broken prematurely. After several such transfers, a cache might have received several ranges of the same entity-body.
If a cache has a stored non-empty set of subranges for an entity, 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," January 2008.)).

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.
For entity-header fields, both sender and recipient refer to either the client or the server, depending on who sends and who receives the entity.

5.1. Accept-Ranges

TOC

The Accept-Ranges response-header field allows the server to indicate its acceptance of range requests for a resource:

```
Accept-Ranges     = "Accept-Ranges" ":" 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 byte-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 entity-header is sent with a partial entity-body to specify where in the full entity-body the partial body should be applied. Range units are defined in Section 2 (Range Units).

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

content-range-spec      = byte-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
```

The header SHOULD indicate the total length of the full entity-body, 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.

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 entity 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). See Appendix B.1 (Changes from RFC 2068) for a compatibility issue.
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 byte-ranges in a single request.
When a client requests multiple byte-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 entity).
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 an entity in its cache, and wishes to have an up-to-date copy of the entire entity 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 entity has been modified, the client would then have to make a second request to obtain the entire current entity-body.
The If-Range header allows a client to "short-circuit" the second request. Informally, its meaning is `if the entity is unchanged, send me the part(s) that I am missing; otherwise, send me the entire new entity'.

```
If-Range = "If-Range" ":" ( entity-tag | HTTP-date )
```

If the client has no entity tag for an entity, 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 entity tag for the entity, then the server SHOULD provide the specified sub-range of the entity using a 206 (Partial Content) response. If the entity tag does not match, then the server SHOULD return the entire entity using a 200 (OK) response.

5.4. Range

TOC

TOC

5.4.1. Byte Ranges

Since all HTTP entities are represented in HTTP messages as sequences of bytes, the concept of a byte range is meaningful for any HTTP entity. (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 entity-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 entity.

```
ranges-specifier = byte-ranges-specifier
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 entity-body, last-byte-pos is taken to be equal to one less than the current length of the entity-body in bytes.
By its choice of last-byte-pos, a client can limit the number of bytes retrieved without knowing the size of the entity.

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

A suffix-byte-range-spec is used to specify the suffix of the entity-body, of a length given by the suffix-length value. (That is, this form specifies the last N bytes of an entity-body.) If the entity is shorter than the specified suffix-length, the entire entity-body 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 entity-body, 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 status of 416 (Requested range not satisfiable). Otherwise, the server SHOULD return

a response with a status of 206 (Partial Content) containing the satisfiable ranges of the entity-body.
Examples of byte-ranges-specifier values (assuming an entity-body 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

HTTP retrieval requests using conditional or unconditional GET methods MAY request one or more sub-ranges of the entity, instead of the entire entity, using the Range request header, which applies to the entity returned as the result of the request:

```
Range = "Range" ":" ranges-specifier
```

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 entities. If the server supports the Range header and the specified range or ranges are appropriate for the entity:

- 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 entity 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

TBD.

---

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,” January 2008.).

---

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-01 (work in progress), January 2008.
[Part3]	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 3: Message Payload and Content Negotiation," draft-ietf-httpbis-p3-payload-01 (work in progress), January 2008.
[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-01 (work in progress), January 2008.
[Part6]	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 6: Caching," draft-ietf-httpbis-p6-cache-01 (work in progress), January 2008.
[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.

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.

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. The media type for this purpose is called "multipart/byteranges". The multipart/byteranges media type includes two 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.

> **Media Type name:** multipart
>
> **Media subtype name:** byteranges
>
> **Required parameters:** boundary
>
> **Optional parameters:** none
>
> **Encoding considerations:** only "7bit", "8bit", or "binary" are permitted
>
> **Security considerations:** none

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--
```

Notes:

1. Additional CRLFs may precede the first boundary string in the entity.

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 2068

TOC

Transfer-coding and message lengths all interact in ways that required fixing exactly when chunked encoding is used (to allow for transfer encoding that may not be self delimiting); it was important to straighten out exactly how message lengths are computed. (Section 5.2 (Content-Range), see also [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,” January 2008.), [Part3] (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 3: Message Payload and Content Negotiation,” January 2008.) and [Part6] (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 6: Caching,” January 2008.))

There are situations where a server (especially a proxy) does not know the full length of a response but is capable of serving a byterange request. We therefore need a mechanism to allow byteranges with a content-range not indicating the full length of the message. (Section 5.2 (Content-Range))

Range request responses would become very verbose if all meta-data were always returned; by allowing the server to only send needed headers in a 206 response, this problem can be avoided. (Section 3.1 (206 Partial Content) and 5.3 (If-Range))

Fix problem with unsatisfiable range requests; there are two cases: syntactic problems, and range doesn't exist in the document. The 416 status code was needed to resolve this ambiguity needed to indicate an error for a byte range request that falls outside of the actual contents of a document. (Section 3.2 (416 Requested Range Not Satisfiable), 5.2 (Content-Range))

B.2. 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))

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

TOC

C.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.).

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

TOC

Closed issues:

- http://www3.tools.ietf.org/wg/httpbis/trac/ticket/18: "Cache validators in 206 responses" (http://purl.org/NET/http-errata#ifrange206)
- http://www3.tools.ietf.org/wg/httpbis/trac/ticket/35: "Normative and Informative references"
- http://www3.tools.ietf.org/wg/httpbis/trac/ticket/86: "Normative up-to-date references"

Index

TOC

2	
	206 Partial Content (status code)
4	
	416 Requested Range Not Satisfiable (status code)
A	
	Accept-Ranges header
C	
	Content-Range header
G	
	Grammar
	Accept-Ranges
	acceptable-ranges
	byte-content-range-spec

	byte-range-resp-spec
	byte-range-set
	byte-range-spec
	byte-ranges-specifier
	bytes-unit
	Content-Range
	content-range-spec
	first-byte-pos
	If-Range
	instance-length
	last-byte-pos
	other-range-unit
	Range
	range-unit
	ranges-specifier
	suffix-byte-range-spec
	suffix-length
H	
	Headers
	Accept-Ranges
	Content-Range
	If-Range
	Range
I	
	If-Range header
M	
	Media Type
	multipart/byteranges
	multipart/x-byteranges
	multipart/byteranges Media Type
	multipart/x-byteranges Media Type
R	
	Range header
S	
	Status Codes
	206 Partial Content
	416 Requested Range Not Satisfiable

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
	One Laptop per Child
	21 Oak Knoll Road
	Carlisle, MA 01741
	USA
Email:	jg@laptop.org
URI:	http://www.laptop.org/
	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/

Full Copyright Statement

TOC

Copyright © The IETF Trust (2008).
This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.
This document and the information contained herein are provided on an “AS IS” basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the

procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.