

## Hypertext Transfer Protocol (HTTP/1.1): Range Requests

문서 최종 수정일	2020-05-26
원문 복사일	2020-05-04
번역 및 정리	이병록(roka88)
이메일	roka88.dev@gmail.com

PROPOSED STANDARD

[Errata Exist](#)

Internet Engineering Task Force (IETF)

Request for Comments: 7233

Obsoletes: [2616](#)

Category: Standards Track

ISSN: 2070-1721

R. Fielding, Ed.

Adobe

Y. Lafon, Ed.

W3C

J. Reschke, Ed.

greenbytes

June 2014

## Hypertext Transfer Protocol (HTTP/1.1): Range Requests

### Abstract

The Hypertext Transfer Protocol (HTTP) is a stateless application-level protocol for distributed, collaborative, hypertext information systems. This document defines range requests and the rules for constructing and combining responses to those requests.

하이퍼텍스트 전송 프로토콜(HTTP)은 분산, 협업, 하이퍼텍스트 정보 시스템을 위한 상태 비저장 애플리케이션 레벨 프로토콜이다. 이 문서는 범위 요청과 해당 요청에 대한 응답을 구성하고 결합하기 위한 규칙을 정의한다.

### Status of This Memo

This is an Internet Standards Track document.

이것은 인터넷 표준 추적 문서이다.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in [Section 2 of RFC 5741](#).

이 문서는 Internet Engineering Task Force(IETF)의 제품이다. 문서는 IETF 공동체의 합의를 나타낸다. 문서는 공개 검토를 받아왔으며 Internet Engineering Starting Group (IESG)에 의해 발행 승인을 받았다. 인터넷 표준의 추가 정보는 [RFC 5741 Section 2](#)에서 확인할 수 있다.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <http://www.rfc-editor.org/info/rfc7233>.

이 문서에 대한 현재 상태 정보는 정오표와 피드백을 어떻게 제공하는 방법은 <http://www.rfc-editor.org/info/rfc7233>에서 얻을 수 있다.

## Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

2014 IETF 트러스트 및 문서 작성자로 식별된 사람.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

무단 전재 금지 이 문서는 [BCP78](http://trustee.ietf.org/license-info) 및 IETF 문서와 관련된 IETF 트러스트의 법적 조항(<http://trustee.ietf.org/license-info>)는 본 문서의 발행일에 유효하다. 이 문서는 본 문서와 관련된 귀하의 권리와 제한 사항을 설명하므로 주의 깊게 검토해야 한다. 이 문서에서 추출된 코드 구성 요소는 신뢰 법률 조항의 섹션 4.e 에 설명된 대로 간소화된 BSD 라이선스 텍스트를 포함해야 하며, Simplified BSD 라이선스에 설명된 대로 보증 없이 제공된다.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

이 문서는 2008년 11월 10일 이전에 공개되거나 공개된 IETF 문서 또는 IETF 계약에서 나온 자료를 포함할 수 있다. 이 자료의 일부에서 저작권을 관리하는 당사자는 IETF 표준 프로세스 밖에서 이러한 자료의 변경을 허용할 권한을 IETF 트러스트에 부여하지 않았을 수 있다. 이러한 자료의 저작권을 관리하는 개인으로부터 적절한 라이선스를 획득하지 않는 한, 이 문서는 IETF 표준 프로세스 외부에서 수정될 수 없으며, 이 문서의 파생 저작물은 RFC로 발행하거나 이를 다른 언어로 변환하는 것을 제외하고는 IETF 표준 프로세스 외부에서 만들어지지 않을 수 있다

## Table of Contents

1. Introduction
  - 1.1 Conformance and Error Handling
  - 1.2 Syntax Notation
2. Range Units
  - 2.1 Byte Ranges
  - 2.2 Other Range Units
  - 2.3 Accept-Ranges
3. Range Requests
  - 3.1 Range
  - 3.2 If-Range
4. Responses to a Range Request
  - 4.1 206 Partial Content
  - 4.2 Content-Range
  - 4.3 Combining Ranges

- 4.4 416 Range Not Satisfiable
- 5. IANA Considerations
  - 5.1 Range Unit Registry
    - 5.1.1 Procedure
    - 5.1.2 Registrations
  - 5.2 Status Code Registration
  - 5.3 Header Field Registration
  - 5.4 Internet Media Type Registration
    - 5.4.1 Internet Media Type multipart/byteranges
- 6. Security Considerations
  - 6.1 Denial-of-Service Attacks Using Range
- 7. Acknowledgements
- 8. Reference
  - 8.1 Normative References
  - 8.2 Informative References
- Appendix A. Internet Media Type multipart/byteranges
- Appendix B. Changes from [RFC 2616](#)
- Appendix C. Imported ABNF
- Appendix D Collected ABNF
- Index

## 1. Introduction

Hypertext Transfer Protocol (HTTP) clients often encounter interrupted data transfers as a result of canceled requests or dropped connections. When a client 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. Likewise, devices with limited local storage might benefit from being able to request only a subset of a larger representation, such as a single page of a very large document, or the dimensions of an embedded image.

Hypertext Transfer Protocol (HTTP) 클라이언트는 요청이 취소되거나 연결이 끊어진 결과로 데이터 전송이 중단되는 경우가 많다. 클라이언트가 부분적인 표현을 저장한 경우, 전체 표현을 전송하는 것 보다 후속 요청으로 그 표현의 나머지를 요청하는 것이 바람직하다. 마찬가지로, 로컬 스토리지가 제한된 장치는 매우 큰 문서의 단일 페이지 또는 내장된 이미지의 치수와 같은 더 큰 표현 중 일부만 요청할 수 있다는 이점을 얻을 수 있다.

This document defines HTTP/1.1 range requests, partial responses, and the multipart/byteranges media type. Range requests are an OPTIONAL feature of

HTTP, designed so that recipients not implementing this feature (or not supporting it for the target resource) 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 caches that might not implement the feature.

본 문서는 HTTP/1.1 범위 요청, 부분 응답 및 multipart/byteranges 미디어 타입을 정의한다. 범위 요청은 HTTP의 선택적 기능으로, 이 기능을 구현하지 않거나 대상 리소스에 대해 지원하지 않는 수신자가 상호운용성에 영향을 주지 않고 정상적인 GET 요청인 것처럼 응답할 수 있도록 설계되었다. 부분 응답은 그 특성을 구현하지 못할 수도 있는 캐시에 의해 전체 응답으로 오인되지 않도록 구별된 상태 코드로 표시된다.

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

범위 요청 메커니즘은 확장 가능한 범위 유형을 허용하도록 설계되었지만, 이 명세는 바이트 범위에 대한 요청만 정의한다.

## 1.1. Conformance and Error Handling

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 [\[RFC2119\]](#).

Conformance criteria and considerations regarding error handling are defined in [Section 2.5 of \[RFC7230\]](#).

## 1.2. Syntax Notation

This specification uses the Augmented Backus-Naur Form (ABNF) notation of [\[RFC5234\]](#) with a list extension, defined in [Section 7 of \[RFC7230\]](#), that allows for compact definition of comma-separated lists using a '#' operator (similar to how the '\*' operator indicates repetition). [Appendix C](#) describes rules imported from other documents. [Appendix D](#) shows the collected grammar with all list operators expanded to standard ABNF notation.

## 2. Range Units

A representation can be partitioned into subranges according to various structural units, depending on the structure inherent in the representation's media type. This "range unit" is used in the Accept-Ranges ([Section 2.3](#)) response header field to advertise support for range requests, the Range ([Section 3.1](#)) request header field to delineate the parts of a representation that are requested, and the Content-Range ([Section 4.2](#)) payload header field to describe which part of a representation is being transferred.

표현은 표현 미디어 타입에 내재된 구조에 따라 다양한 구조 단위에 따라 하위 영역으로 분할될 수 있다. 이 "range unit"는 범위 요청에 대한 지원을 알리기 위해 Accept-Ranges (Section 2.3) 응답 헤더 필드에서 사용되며, 요청된 표현 부분을 기술하기 위해 Range 요청 헤더 필드와(Section 3.1), 그리고 표현 중 어느 부분이 전달되고 있는지를 기술하기 위해 Content-Range (Section 4.2) 페이로드 헤더 필드에서 사용된다.

range-unit = bytes-unit / other-range-unit

### 2.1. Byte Ranges

Since representation data is transferred in payloads as a sequence of octets, a byte range is a meaningful substructure for any representation transferable over HTTP ([Section 3 of \[RFC7231\]](#)). The "bytes" range unit is defined for expressing subranges of the data's octet sequence.

표현 데이터는 octets으로 페이로드로 전송되기 때문에, 바이트 범위는 HTTP를 통해 전송할 수 있는 모든 표현에 대한 의미 있는 하부 구조다([RFC7231]의 Section 3). "bytes" 범위 단위는 데이터의 octet 시퀀스의 하위 범위를 표현하기 위해 정의된다.

bytes-unit = "bytes"

A byte-range request can specify a single range of bytes or a set of ranges within a single representation.

byte-range 요청은 단일 표시 범위 내에서 단일 바이트 범위 또는 범위 집합을 지정할 수 있다.

```
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.

byte-range-spec의 first-byte-pos 값은 범위의 첫 번째 바이트의 byte-offset을 제공한다. last-byte-pos 값은 범위의 마지막 바이트의 byte-offset을 제공한다. 즉, 지정된 바이트 위치는 포함된다. 바이트 오프셋은 0에서 시작한다.

Examples of byte-ranges-specifier values:

byte-ranges-specifier값의 예시:

- o The first 500 bytes (byte offsets 0-499, inclusive):
- o 첫 번째 500 바이트 (바이트 오프셋 0-499, 포함)

```
bytes=0-499
```

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



- o 두 번째 500 바이트 (바이트 오프셋 500-999, 포함)

bytes=500-999

A byte-range-spec is invalid if the last-byte-pos value is present and less than the first-byte-pos.

byte-range-spec은 last-byte-pos 값이 있고 first--byte-pos보다 작으면 유효하지 않다.

A client can limit the number of bytes requested without knowing the size of the selected representation. If the last-byte-pos value is absent, or if the value is greater than or equal to the current length of the representation data, the byte range is interpreted as the remainder of the representation (i.e., the server replaces the value of last-byte-pos with a value that is one less than the current length of the selected representation).

클라이언트는 선택된 표현의 크기를 아는 것 없이 요청된 바이트 수를 제한할 수 있다. last-byte-pos 값이 없거나 값이 표현 데이터의 현재 길이보다 크거나 같으면 바이트 범위는 표현의 나머지로 해석된다(i.e., 서버는 last-byte-pos의 값을 선택된 표현의 현재 길이보다 하나 작은 값으로 대체한다).

A client can request the last N bytes of the selected representation using a suffix-byte-range-spec.

클라이언트는 suffix-byte-range-spec을 사용하여 선택된 표현의 마지막 N바이트를 요청할 수 있다.

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

If the selected representation is shorter than the specified suffix-length, the entire representation is used.

선택된 표현이 지정된 suffix-length보다 짧으면 전체 표현이 사용된다.

Additional examples, assuming a representation of length 10000:

추가적인 예제로, 길이 10000의 예제가 있다고 가정하면:

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

- o 마지막 500 바이트 (바이트 오프셋 9500-9999, 포함)

bytes=-500

Or:

bytes=9500-

- o The first and last bytes only (bytes 0 and 9999):

- o 첫 번째와 마지막 바이트만 (바이트 0 그리고 9999)

bytes=0-0,-1

- o Other valid (but not canonical) specifications of the second 500 bytes (byte offsets 500-999, inclusive):

- o 나머지 유효한(표준은 아님) 두 번째 500 바이트의 명세 (바이트 오프셋 500-999, 포함)

bytes=500-600,601-999

bytes=500-700,601-999

If a valid byte-range-set includes at least one byte-range-spec with a first-byte-pos that 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.

유효한 바이트 범위 집합이 표현의 현재 길이보다 작은 first-byte-pos와 적어도 하나의 byte-range-spec 또는 0이 아닌 suffix-length와 suffix-byte-range-spec을 포함하는 경우, 바이트 byte-range-set은 충족된다. 그렇지 않으면 byte-range-set이 충족되지 못한다.

In the byte-range syntax, first-byte-pos, last-byte-pos, and suffix-length are expressed as decimal number of octets. Since there is no predefined limit to the length of a payload, recipients MUST anticipate potentially large decimal numerals and prevent parsing errors due to integer conversion overflows.

byte-range 구문에서 first-byte-pos, last-byte-pos, suffix-length는 십진수 octet으로 표현된다. 페이로드 길이에 대한 사전 정의된 제한이 없으므로 수신자는 잠재적으로 큰 십진수를 예상해야 하며 정수 변환 오버플로로 인한 구문 분석 오류를 방지해야 한다.(MUST)

## 2.2. Other Range Units

Range units are intended to be extensible. New range units ought to be registered with IANA, as defined in [Section 5.1](#).

범위 단위는 확장 가능하도록 설계되어 있다. 새 범위 단위는 Section 5.1에 정의된 대로 IANA에 등록해야 한다.

other-range-unit = token

## 2.3. Accept-Ranges

The "Accept-Ranges" header field allows a server to indicate that it supports range requests for the target resource.

"Accept-Ranges" 헤더 필드는 서버가 대상 리소스에 대한 범위 요청을 지원함을 나타낼 수 있도록 한다.

Accept-Ranges = acceptable-ranges  
acceptable-ranges = 1#range-unit / "none"

An origin server that supports byte-range requests for a given target resource MAY send

Accept-Ranges: bytes

to indicate what range units are supported. A client MAY generate range requests without having received this header field for the resource involved. Range units are defined in [Section 2](#).

지정된 대상 리소스에 대한 바이트 범위 요청을 지원하는 원서버가 지원되는 범위 단위를 표시하기 위해 Accept-Ranges: bytes를 보낼 수 있다.(MAY) 클라이언트는 관련된 리소스에 대한 이 헤더 필드를 수신하지 않고 범위 요청을 생성할 수 있다.(MAY) 범위 단위는 Section 2에 정의되어 있다.

A server that does not support any kind of range request for the target resource MAY send

Accept-Ranges: none

to advise the client not to attempt a range request.

대상 리소스에 대한 어떤 종류의 범위 요청도 지원하지 않는 서버가 클라이언트에게 범위 요청을 시도하지 말 것을 권고하기 위해 Accept-Ranges: none를 보낼 수 있다.(MAY)

### 3. Range Requests

#### 3.1. Range

The "Range" header field on a GET request modifies the method semantics to request transfer of only one or more subranges of the selected representation data, rather than the entire selected representation data.

GET 요청의 "Range" 헤더 필드는 선택된 전체 표현 데이터가 아닌 선택된 표현 데이터의 하위 영역 하나 이상의 전송만 요청하도록 메서드 의미를 수정한다.

Range = byte-ranges-specifier / other-ranges-specifier  
other-ranges-specifier = other-range-unit "=" other-range-set  
other-range-set = 1\*VCHAR

A server MAY ignore the Range header field. However, origin servers and intermediate caches ought to support byte ranges when possible, since Range supports efficient recovery from partially failed transfers and partial retrieval of large representations. A server MUST ignore a Range header field received with a request method other than GET.

서버는 Range 헤더 필드를 무시할 수 있다.(MAY) 단, 원서버와 중간 캐시는 가능한 경우 바이트 범위를 지원해야 한다. Range는 부분적으로 실패한 전송으로부터 효율적인 복구와 대규모 표현의 부분 검색을 지원하기 때문이다. 서버는 GET 이외의 요청 메서드로 수신된 범위 헤더 필드를 무시해야 한다.(MUST)

An origin server MUST ignore a Range header field that contains a range unit it does not understand. A proxy MAY discard a Range header field that contains a range unit it does not understand.

원서버는 이해하지 못하는 범위 단위가 포함된 범위 헤더 필드를 무시해야 한다.(MUST) 프락시는 이해하지 못하는 범위 단위가 포함된 Range 헤더 필드를 폐기할 수 있다.(MAY)

A server that supports range requests MAY ignore or reject a Range header field that consists of more than two overlapping ranges, or a set of many small ranges that are not listed in ascending order, since both are indications of either a broken client or a deliberate denial-of-service attack ([Section 6.1](#)). A client SHOULD NOT request multiple ranges that are inherently less efficient to process and transfer than a single range that encompasses the same data.

범위 요청을 지원하는 서버는 두 개 이상의 겹치는 범위로 구성되거나, 또는 오름차순으로 나열되지 않은 많은 작은 범위의 집합이면, 고장난 클라이언트 또는 의도적인 denial-of-service 공격으로 나타내기 때문에 Range 헤더 필드를 무시하거나 거부할 수 있다.(MAY) (Section 6.1). 클라이언트는 동일한 데이터를 포함하는 단일 범위보다 처리 및 전송 효율성이 본질적으로 낮은 여러 범위를 요청해서는 안 된다.(SHOULD NOT)

A client that is requesting multiple ranges SHOULD list those ranges in ascending order (the order in which they would typically be received in a complete representation) unless there is a specific need to request a later part earlier. For example, a user agent processing a large representation with an internal catalog of parts might need to request later parts first, particularly if the representation consists of pages stored in reverse order and the user agent wishes to transfer one page at a time.

다중 범위를 요청하는 클라이언트는 더 이전 부분을 요청할 특별한 필요가 없는 한 그러한 범위를 오름차순(일반적으로 완전한 표현으로 수신되는 순서)으로 나열해야 한다.(SHOULD) 예를 들어, 부분의 내부 카탈로그로 대량의 표현을 처리하는 사용자 에이전트는, 특히 표현이 역순으로 저장된 페이지로 구성되어 있고 사용자 에이전트가 한 번에 한 페이지씩 전송하기를 원하는 경우, 이후 부분을 먼저 요청해야 할 수 있다.

The Range header field is evaluated after evaluating the precondition header fields defined in [\[RFC7232\]](#), and only if the result in absence of the Range header field would be a 200 (OK) response. In other words, Range is ignored when a conditional GET would result in a 304 (Not Modified) response.

Range 헤더 필드는 [\[RFC7232\]](#)에서 정의한 전제조건 헤더 필드를 평가한 후 평가되며, Range 헤더 필드가 없는 결과가 200 (OK) 응답일 경우에만 평가된다. 즉, 조건부 GET가 304 (Not Modified) 응답을 초래할 경우 Range는 무시된다.

The If-Range header field ([Section 3.2](#)) can be used as a precondition to applying the Range header field.

If-Range 헤더 필드(Section 3.2)는 Range 헤더 필드를 적용하기 위한 전제 조건으로 사용할 수 있다.

If all of the preconditions are true, the server supports the Range header field for the target resource, and the specified range(s) are valid and satisfiable (as defined in [Section 2.1](#)), the server SHOULD send a 206 (Partial Content) response with a

payload containing one or more partial representations that correspond to the satisfiable ranges requested, as defined in [Section 4](#).

모든 전제 조건이 참인 경우, 서버는 대상 리소스에 대한 Range 헤더 필드를 지원하며, 지정된 범위가 유효하고 충족가능한 경우(Section 2.1에서 정의) 서버는 Section 4에 정의된 바와 같이 요청된 충족가능한 범위에 해당하는 하나 이상의 부분적 표현을 포함하는 페이로드와 함께 206 (Partial Content) 응답을 보내야 한다.(SHOULD)

If all of the preconditions are true, the server supports the Range header field for the target resource, and the specified range(s) are invalid or unsatisfiable, the server SHOULD send a 416 (Range Not Satisfiable) response.

모든 전제 조건이 참인 경우, 서버는 대상 리소스에 대한 Range 헤더 필드를 지원하며, 지정된 범위가 유효하지 않거나 충족할 수 없는 경우, 서버는 416 (Range Not Satisfiable) 응답을 보내야 한다.(SHOULD)

### 3.2. If-Range

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

클라이언트가 표현의 일부 사본을 가지고 있고 전체 표현에 대한 최신 사본을 갖고자 하는 경우, GET 조건부(If-Unmodified-Since와 If-Match 중 하나 또는 둘 다 사용)의 Range 헤더 필드를 사용할 수 있다. 그러나 표현이 수정되었기 때문에 전제조건이 실패한다면, 클라이언트는 현재 표현 전체를 얻기 위해 두 번째 요청을 해야 할 것이다.

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

"If-Range" 헤더 필드에서는 클라이언트가 두 번째 요청을 "short-circuit"할 수 있다. 비공식적으로, 그것의 의미는 다음과 같다: 만약 표현이 변하지 않는다면, 내가 Range에서 요청하고 있는 부분을 나에게 보내라. 그렇지 않으면, 전체 표현을 나에게 전송해라.

If-Range = entity-tag / HTTP-date

A client MUST NOT generate an If-Range header field in a request that does not contain a Range header field. A server MUST ignore an If-Range header field received in a request that does not contain a Range header field. An origin server MUST ignore an If-Range header field received in a request for a target resource that does not support Range requests.

클라이언트는 Range 헤더 필드가 없는 요청에서 If-Range 헤더 필드를 생성해서는 안 된다. (MUST NOT) 서버는 Range 헤더 필드가 없는 요청으로 수신된 If-Range 헤더 필드를 무시해야 한다. (MUST) 원서버는 범위 요청을 지원하지 않는 대상 리소스에 대한 요청으로 수신된 If-Range 헤더 필드를 무시해야 한다. (MUST)

A client MUST NOT generate an If-Range header field containing an entity-tag that is marked as weak. A client MUST NOT generate an If-Range header field containing an HTTP-date unless the client has no entity-tag for the corresponding representation and the date is a strong validator in the sense defined by [Section 2.2.2 of \[RFC7232\]](#).

클라이언트는 약한 entity-tag가 포함된 If-Range 헤더 필드를 생성해서는 안 된다. (MUST NOT) 클라이언트가 해당 표현에 대한 엔터티 태그가 없고 날짜가 [RFC7232]의 Section 2.2.2에서 정의한 강한 검증자가 아닌 한 클라이언트는 HTTP-date가 포함된 If-Range 헤더 필드를 생성해서는 안 된다. (MUST NOT)

A server that evaluates an If-Range precondition MUST use the strong comparison function when comparing entity-tags ([Section 2.3.2 of \[RFC7232\]](#)) and MUST evaluate the condition as false if an HTTP-date validator is provided that is not a strong validator in the sense defined by [Section 2.2.2 of \[RFC7232\]](#). A valid entity-tag can be distinguished from a valid HTTP-date by examining the first two characters for a DQUOTE.

If-Range 전제조건을 평가하는 서버는 entity-tags ([RFC7232]의 Section 2.3.2)를 비교할 때 강한 비교 함수를 사용해야 하며 [RFC7232]의 Section 2.2.2에 정의한 강한 검증자가 아



닌 HTTP-date 검증자가 제공된 경우 조건을 거짓으로 평가해야 한다. 유효한 entity-tag는 DQUOTE에 대한 처음 두 문자를 검토하여 유효한 HTTP-date와 구별할 수 있다.

If the validator given in the If-Range header field matches the current validator for the selected representation of the target resource, then the server SHOULD process the Range header field as requested. If the validator does not match, the server MUST ignore the Range header field. Note that this comparison by exact match, including when the validator is an HTTP-date, differs from the "earlier than or equal to" comparison used when evaluating an If-Unmodified-Since conditional.

If-Range 헤더 필드에 주어진 검증자가 대상 리소스의 선택된 표현에 대한 현재 검증자와 일치하는 경우, 서버는 요청대로 Range 헤더 필드를 처리해야 한다.(SHOULD) 검증자가 일치하지 않으면 서버는 Range 헤더 필드를 무시해야 한다.(MUST) 검증자가 HTTP-date일 때를 포함하여 정확한 일치에 의한 이 비교는 If-Unmodified-Since 조건부 평가 시 사용되는 "earlier than or equal to" 비교와 다르다는 점에 참고한다.

## 4. Responses to a Range Request

### 4.1. 206 Partial Content

The 206 (Partial Content) status code indicates that the server is successfully fulfilling a range request for the target resource by transferring one or more parts of the selected representation that correspond to the satisfiable ranges found in the request's Range header field ([Section 3.1](#)).

206(Partial Content) 상태 코드는 서버가 요청의 Range 헤더 필드(Section 3.1)에서 찾은 충족가능한 범위에 해당하는 선택된 표현 중 하나 이상을 전송하여 대상 리소스에 대한 범위 요청을 성공적으로 이행하고 있음을 나타낸다.

If a single part is being transferred, the server generating the 206 response MUST generate a Content-Range header field, describing what range of the selected representation is enclosed, and a payload consisting of the range. For example:

단일 부분이 전송되는 경우, 206 응답을 생성하는 서버는 선택된 표현의 범위를 설명하는 Content-Range 헤더 필드와 그 범위로 구성된 페이로드 필드를 생성해야 한다.(MUST) 예를 들어:

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

... 26012 bytes of partial image data ...

If multiple parts are being transferred, the server generating the 206 response MUST generate a "multipart/byteranges" payload, as defined in [Appendix A](#), and a Content-Type header field containing the multipart/byteranges media type and its required boundary parameter. To avoid confusion with single-part responses, a server MUST NOT generate a Content-Range header field in the HTTP header section of a multiple part response (this field will be sent in each part instead).

여러 부분이 전송되는 경우, 206 응답을 생성하는 서버는 Appendix A에 정의된 "multipart/byteranges" 페이로드와 multipart/byteranges 미디어 타입 및 필요한 경계 매개변수를 포함하는 Content-Type 헤더 필드를 생성해야 한다.(MUST) 단일 부분 응답과 혼동을 방지하려면 서버가 여러 부분 응답의 HTTP 헤더 부문에 있는 Content-Range 헤더 필드를 생성해서는 안 된다.(MUST NOT) (대신 이 필드는 각 부분에서 전송된다).

Within the header area of each body part in the multipart payload, the server MUST generate a Content-Range header field corresponding to the range being enclosed in that body part. If the selected representation would have had a Content-Type header field in a 200 (OK) response, the server SHOULD generate that same Content-Type field in the header area of each body part. For example:

멀티파트 페이로드의 각 본문 부분의 헤더 영역 내에서, 서버는 본문 부분에 동봉되는 범위에 해당하는 Content-Range 헤더 필드를 생성해야 한다.(MUST) 선택된 표현에 200 (OK) 응답의 Content-Type 헤더 필드가 있을 경우, 서버는 각 본문 부분의 헤더 영역에 동일한 Content-Type 필드를 생성해야 한다.(SHOULD) 예를 들어:

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-Length: 1741

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--

When multiple ranges are requested, a server MAY coalesce any of the ranges that overlap, or that are separated by a gap that is smaller than the overhead of sending multiple parts, regardless of the order in which the corresponding byte-range-spec appeared in the received Range header field. Since the typical overhead between parts of a multipart/byteranges payload is around 80 bytes, depending on the selected representation's media type and the chosen boundary parameter length, it can be less efficient to transfer many small disjoint parts than it is to transfer the entire selected representation.

복수 범위가 요청될 경우, 서버는 수신된 Range 헤더 필드에 해당 byte-range-spec이 나타난 순서에 관계없이, 겹치거나 복수의 부분을 보내는 오버헤드보다 작은 갭으로 분리되는 모든 범위를 통합할 수 있다.(MAY) multipart/byteranges 페이로드의 부분들 사이의 일반적인 오버헤드는 약 80 바이트이므로, 선택된 표현의 미디어 타입과 선택한 경계 매개변수 길이에 따라, 선택된 표현 전체를 전달하는 것보다 많은 작은 해체된 부분을 전송하는 것이 덜 효율적일 수 있다.

A server MUST NOT generate a multipart response to a request for a single range, since a client that does not request multiple parts might not support multipart responses. However, a server MAY generate a multipart/byteranges payload with only a single body part if multiple ranges were requested and only one range was found to be satisfiable or only one range remained after coalescing. A client that cannot process a multipart/byteranges response MUST NOT generate a request that asks for multiple ranges.

복수 부분을 요청하지 않는 클라이언트는 멀티파트 응답을 지원하지 않을 수 있으므로 서버는 단일 범위에 대한 요청에 대한 멀티파트 응답을 생성해서는 안 된다.(MUST NOT) 그러나, 복수

의 범위가 요청되어 하나의 범위만 만족되거나 병합이 끝난 후에도 하나의 범위만 남아 있는 경우, 서버는 하나의 본문 부분만 가진 multipart/byteranges 페이로드를 생성할 수 있다.(MAY) multipart/byteranges 응답을 처리할 수 없는 클라이언트는 여러 범위를 요청하는 요청을 생성해서는 안 된다.(MUST NOT)

When a multipart response payload is generated, the server SHOULD send the parts in the same order that the corresponding byte-range-spec appeared in the received Range header field, excluding those ranges that were deemed unsatisfiable or that were coalesced into other ranges. A client that receives a multipart response MUST inspect the Content-Range header field present in each body part in order to determine which range is contained in that body part; a client cannot rely on receiving the same ranges that it requested, nor the same order that it requested.

멀티파트 응답 페이로드 발생 시, 서버는 수신된 Range 헤더 필드에 나타난 해당 byte-range-spec과 동일한 순서로 부분을 전송해야 하며,(SHOULD) 불충족하거나 다른 범위로 결합된 범위를 제외한다. 멀티파트 응답을 수신한 클라이언트는 본문 부분에 어떤 범위가 포함되어 있는지 결정하기 위해 각 본문 부분에 존재하는 Content-Range 헤더 필드를 검사해야 한다.(MUST) 클라이언트는 자신이 요청한 범위 또는 요청했던 순서를 수신하는 데 의존할 수 없다.

When a 206 response is generated, the server MUST generate the following header fields, in addition to those required above, if the field would have been sent in a 200 (OK) response to the same request: Date, Cache-Control, ETag, Expires, Content-Location, and Vary.

206 응답이 생성되었을 때, 서버는 위의 요구 사항 외에, 200(OK) 응답에서 동일한 요청에 필드가 전송되었을 경우, 반드시 다음과 같은 헤더 필드를 생성해야 한다.(MUST): Date, Cache-Control, Etag, Expires, Content-Location 및 Vary

If a 206 is generated in response to a request with an If-Range header field, the sender SHOULD NOT generate other representation header fields beyond those required above, because the client is understood to already have a prior response containing those header fields. Otherwise, the sender MUST generate all of the representation header fields that would have been sent in a 200 (OK) response to the same request.

If-Range 헤더 필드가 있는 요청에 따라 206 응답이 생성되는 경우, 클라이언트는 이미 헤더 필드를 포함하는 사전 응답을 가지고 있는 것으로 이해되므로, 발신자는 위에서 요구한 필드를 초과하는 다른 표현 헤더 필드를 생성해서는 안 된다.(SHOULD NOT) 그렇지 않은 경우, 발신자는 동일한 요청에 대한 200 (OK) 응답으로 전송되었을 모든 표현 헤더 필드를 생성해야 한다.(MUST)

A 206 response is cacheable by default; i.e., unless otherwise indicated by explicit cache controls (see [Section 4.2.2 of \[RFC7234\]](#)).

206 응답은 기본적으로 캐시할 수 있다. 즉, 명시적 캐시 제어에 의해 달리 명시되지 않는 한 ([RFC7234]의 Section 4.2.2 참조).

## 4.2. Content-Range

The "Content-Range" header field is sent in a single part 206 (Partial Content) response to indicate the partial range of the selected representation enclosed as the message payload, sent in each part of a multipart 206 response to indicate the range enclosed within each body part, and sent in 416 (Range Not Satisfiable) responses to provide information about the selected representation.

"Content-Range" 헤더 필드는 메시지 페이로드로 동봉된 선택된 표현의 부분 범위를 나타내기 위해 단일 부분 206(Partial Content) 응답으로 전송되며, 각 본문 부분 내에 동봉된 범위를 나타내기 위해 멀티파트 206 응답의 각 부분으로 전송되며, 선택된 표현에 대한 정보를 제공하기 위해 416(Range Not Satisfiable) 응답으로 전송한다.

Content-Range = byte-content-range  
/ other-content-range

byte-content-range = bytes-unit SP  
( byte-range-resp / unsatisfied-range )

byte-range-resp = byte-range "/" ( complete-length / "\*" )

byte-range = first-byte-pos "-" last-byte-pos

unsatisfied-range = "\*" / complete-length

complete-length = 1 \* DIGIT

other-content-range = other-range-unit SP other-range-resp

other-range-resp = \*CHAR

If a 206 (Partial Content) response contains a Content-Range header field with a range unit ([Section 2](#)) that the recipient does not understand, the recipient MUST NOT attempt to recombine it with a stored representation. A proxy that receives such a message SHOULD forward it downstream.

206(Partial Content) 응답에 수신자가 이해하지 못하는 범위 단위(Section 2)가 있는 Content-Range 헤더 필드가 포함된 경우, 수신자는 저장된 표현으로 재결합하지 않아야 한다. 이러한 메시지를 수신하는 프락시는 해당 메시지를 다운스트림으로 전달해야 한다.(SHOULD)

For byte ranges, a sender SHOULD indicate the complete length of the representation from which the range has been extracted, unless the complete length is unknown or difficult to determine. An asterisk character ("\*") in place of the complete-length indicates that the representation length was unknown when the header field was generated.

바이트 범위의 경우, 송신자는 전체 길이를 알 수 없거나 결정하기 어려운 경우가 아니라면, 범위가 추출된 표현의 전체 길이를 표시해야 한다.(SHOULD) 전체 길이 대신 별표 문자("\*")는 헤더 필드가 생성되었을 때 표현 길이를 알 수 없음을 나타낸다.

The following example illustrates when the complete length of the selected representation is known by the sender to be 1234 bytes:

다음 예는 발신자에 의해 선택된 표현의 전체 길이를 1234바이트로 알고 있는 경우를 예시한다:

Content-Range: bytes 42-1233/1234

and this second example illustrates when the complete length is unknown:

그리고 두 번째 예시는 전체 길이를 모를 경우를 예시한다:

Content-Range: bytes 42-1233/\*

A Content-Range field value is invalid if it contains a byte-range-resp that has a last-byte-pos value less than its first-byte-pos value, or a complete-length value less than or equal to its last-byte-pos value. The recipient of an invalid Content-Range MUST NOT attempt to recombine the received content with a stored representation.

Content-Range 필드 값은 first-byte-pos 값보다 작은 last-byte-pos를 가지거나 또는 last-byte-pos 값보다 작거나 같은 complete-length 값을 가진 byte-range-resp를 포함하는 경우 유효하지 않다. 유효하지 않은 Content-Range의 수신자는 수신된 콘텐츠를 저장된 표현과 재결합하지 않아야 한다.(MUST NOT)

A server generating a 416 (Range Not Satisfiable) response to a byte-range request SHOULD send a Content-Range header field with an unsatisfied-range value, as in the following example:

byte-range 요청에 대해 416(Range Not Satisfiable) 응답을 생성하는 서버는 다음 예와 같이 충족되지 않은 범위 값의 Content-Range 헤더 필드를 전송해야 한다.(SHOULD)

```
Content-Range: bytes */1234
```

The complete-length in a 416 response indicates the current length of the selected representation.

416 응답의 전체 길이는 선택된 표현의 현재 길이를 나타낸다.

The Content-Range header field has no meaning for status codes that do not explicitly describe its semantic. For this specification, only the 206 (Partial Content) and 416 (Range Not Satisfiable) status codes describe a meaning for Content-Range.

Content-Range 헤더 필드는 그 의미론을 명시적으로 설명하지 않는 상태 코드에 대해 의미를 가지고 있지 않다. 이 명세의 경우 206(Partial Content) 및 416(Range Not Satisfiable) 상태 코드만 내용 범위에 대한 의미를 설명한다.

The following are examples of Content-Range values in which the selected representation contains a total of 1234 bytes:

다음은 선택된 표현에 총 1234바이트가 포함된 Content-Range 값의 예다.

- o The first 500 bytes:

- o 첫 번째 500 바이트

Content-Range: bytes 0-499/1234

- o The second 500 bytes:

- o 두 번째 500 바이트

Content-Range: bytes 500-999/1234

- o All except for the first 500 bytes:

- o 첫 번째 500 바이트를 제외한 모든 것(나머지)

Content-Range: bytes 500-1233/1234

- o The last 500 bytes:

- o 마지막 500 바이트

Content-Range: bytes 734-1233/1234



### 4.3. Combining Ranges

A response might transfer only a subrange of a representation if the connection closed prematurely or if the request used one or more Range specifications. After several such transfers, a client might have received several ranges of the same representation. These ranges can only be safely combined if they all have in common the same strong validator ([Section 2.1 of \[RFC7232\]](#)).

커넥션이 일찍 닫히거나 요청이 하나 이상의 Range 명세를 사용한 경우, 응답은 표현의 하위 범위만 전송할 수 있다. 그러한 전송이 몇 차례 있는 후, 클라이언트는 동일한 표현의 몇 가지 범위를 받았을 수 있다. 이러한 범위는 모두 동일한 강한 검증자를 공통으로 가지고 있는 경우에만 안전하게 결합할 수 있다.([RFC7232]의 Section 2.1)

A client that has received multiple partial responses to GET requests on a target resource MAY combine those responses into a larger continuous range if they share the same strong validator.

대상 리소스의 GET 요청에 대해 여러 부분 응답을 받은 클라이언트는 동일한 강한 검증자를 공유하는 경우 이러한 응답을 더 큰 연속 범위로 결합할 수 있다.(MAY)

If the most recent response is an incomplete 200 (OK) response, then the header fields of that response are used for any combined response and replace those of the matching stored responses.

가장 최근의 응답이 불완전한 200(OK) 응답인 경우, 해당 응답의 헤더 필드는 결합된 응답에 사용되며 일치하는 저장된 응답의 헤더 필드를 대체한다.

If the most recent response is a 206 (Partial Content) response and at least one of the matching stored responses is a 200 (OK), then the combined response header fields consist of the most recent 200 response's header fields. If all of the matching stored responses are 206 responses, then the stored response with the most recent header fields is used as the source of header fields for the combined response, except that the client MUST use other header fields provided in the new response, aside from Content-Range, to replace all instances of the corresponding header fields in the stored response.

가장 최근의 응답이 206(Partial Content) 응답이고 일치하는 저장된 응답 중 적어도 하나가 200(OK)인 경우, 결합된 응답 헤더 필드는 가장 최근의 200 응답의 헤더 필드로 구성된다.

일치하는 모든 저장된 응답이 206 응답인 경우, 클라이언트가 저장된 응답에서 해당 헤더 필드의 모든 인스턴스를 대체하기 위해 Content-Range 외에, 새 응답에 제공된 다른 헤더 필드를 사용해야(MUST) 한다는 점을 제외하고, 가장 최근의 헤더 필드가 포함된 저장된 응답은 결합된 응답의 헤더 필드의 소스로 사용된다.

The combined response message body consists of the union of partial content ranges in the new response and each of the selected responses. If the union consists of the entire range of the representation, then the client MUST process the combined response as if it were a complete 200 (OK) response, including a Content-Length header field that reflects the complete length. Otherwise, the client MUST process the set of continuous ranges as one of the following: an incomplete 200 (OK) response if the combined response is a prefix of the representation, a single 206 (Partial Content) response containing a multipart/byteranges body, or multiple 206 (Partial Content) responses, each with one continuous range that is indicated by a Content-Range header field.

결합된 응답 메시지 본문은 새로운 응답의 부분적인 내용 범위의 조합과 선택된 각 응답으로 구성된다. 조합이 전체 표현범위로 구성된 경우, 클라이언트는 결합된 응답을 전체 길이를 반영하는 Content-Length 헤더 필드를 포함하여 완전한 200(OK) 응답인 것처럼 처리해야 한다.(MUST) 그렇지 않으면 클라이언트는 연속 범위 집합을 다음 중 하나로 처리해야 한다:(MUST) 결합된 응답이 표현의 접두사인 경우 불완전한 200(OK) 응답, multipart/byteranges 본문을 포함하는 단일 206(Partial Content) 응답, 또는 복수의 206(Partial Content) 응답, 각각 하나의 Content-Range 헤더 필드로 표시된 연속 범위를 사용하여 처리한다.

#### 4.4. 416 Range Not Satisfiable

The 416 (Range Not Satisfiable) status code indicates that none of the ranges in the request's Range header field ([Section 3.1](#)) overlap the current extent of the selected resource or that the set of ranges requested has been rejected due to invalid ranges or an excessive request of small or overlapping ranges.

416 (Range Not Satisfiable) 상태 코드는 요청의 Range 헤더 필드(Section 3.1)에 있는 어떤 범위도 선택한 리소스의 현재 범위와 겹치지 않거나 잘못된 범위 또는 작은 범위나 중복된 범위의 과도한 요청으로 인해 요청된 범위 집합이 거부되었음을 나타낸다.

For byte ranges, failing to overlap the current extent means that the first-byte-pos of all of the byte-range-spec values were greater than the current length of the selected representation. When this status code is generated in response to a byte-

range request, the sender SHOULD generate a Content-Range header field specifying the current length of the selected representation ([Section 4.2](#)).

바이트 범위의 경우, 현재 범위와 겹치지 않는 것은 모든 byte-range-spec 값의 first-byte-pos가 선택된 표현의 현재 길이보다 크다는 것을 의미한다. byte-range 요청에 대응하여 이상태 코드가 생성되면 발신자는 선택된 표현의 현재 길이를 지정하는 내용 범위 헤더 필드를 생성해야 한다.(SHOULD) (Section 4.2)

For example:

예시로:

```
HTTP/1.1 416 Range Not Satisfiable
Date: Fri, 20 Jan 2012 15:41:54 GMT
Content-Range: bytes */47022
```

Note: Because servers are free to ignore Range, many implementations will simply respond with the entire selected representation in a 200 (OK) response. That is partly because most clients are prepared to receive a 200 (OK) to complete the task (albeit less efficiently) and partly because clients might not stop making an invalid partial request until they have received a complete representation. Thus, clients cannot depend on receiving a 416 (Range Not Satisfiable) response even when it is most appropriate.

참고: 서버들은 Range를 무시하는 것이 자유롭기 때문에, 많은 구현들은 200 (OK) 응답으로 선택된 전체 표현으로 간단히 응답할 것이다. 그것은 부분적으로는 대부분의 클라이언트가 작업을 완료하기 위해 200(OK)을 받을 준비가 되어 있기 때문이며(비록 효율성이 낮음) 부분적으로 클라이언트가 완전한 표현을 받을 때까지 유효하지 않은 부분 요청을 중단하지 않을 수 있기 때문이다. 따라서, 클라이언트는 416 (Range Not Satisfiable) 응답을 가장 적절할 때 수신하는 것에 의존할 수 없다.

## 5. IANA Considerations

### 5.1. Range Unit Registry

The "HTTP Range Unit Registry" defines the namespace for the range unit names and refers to their corresponding specifications. The registry has been created and is now maintained at <<http://www.iana.org/assignments/http-parameters>>.

"HTTP Range Unit Registry"는 범위 단위 이름에 대한 네임스페이스를 정의하고 해당 명세를 참조한다. Registry가 만들어져 현재 <<http://www.iana.org/assignments/http-parameters>>에서 유지되고 있다.

### 5.1.1. Procedure

Registration of an HTTP Range Unit MUST include the following fields:

HTTP Range Unit 등록에는 다음 필드가 포함되어야 한다.(MUST)

- o Name
- o 이름
- o Description
- o 설명
- o Pointer to specification text
- o 명세에 대해 가리키는 것

Values to be added to this namespace require IETF Review (see [\[RFC5226\], Section 4.1](#)).

이 네임스페이스에 추가할 값은 IETF 검토가 필요하다([RFC5226, Section 4.1 참조]).

### 5.1.2. Registrations

The initial range unit registry contains the registrations below:

초기 범위 단위 레지스트리는 아래의 등록을 포함한다.

Range Unit Name	Description	Reference
bytes	a range of octets	<a href="#">Section 2.1</a>
none	reserved as keyword, indicating no ranges are supported	<a href="#">Section 2.3</a>

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

## 5.2. Status Code Registration

The "Hypertext Transfer Protocol (HTTP) Status Code Registry" located at <http://www.iana.org/assignments/http-status-codes> has been updated to include the registrations below:

<http://www.iana.org/assignments/http-status-codes>에 위치한 "Hypertext Transfer Protocol (HTTP) Status Code Registry"는 아래 등록 내용을 포함하도록 업데이트되었다.

Value	Description	Reference
206	Partial Content	<a href="#">Section 4.1</a>
416	Range Not Satisfiable	<a href="#">Section 4.4</a>

## 5.3. Header Field Registration

HTTP header fields are registered within the "Message Headers" registry maintained at <http://www.iana.org/assignments/message-headers/>.

HTTP 헤더 필드는 <http://www.iana.org/assignments/message-headers/>에서 유지되는 "Message Headers" 레지스트리 내에 등록된다.

This document defines the following HTTP header fields, so their associated registry entries have been updated according to the permanent registrations below (see [BCP90]):

이 문서는 다음 HTTP 헤더 필드를 정의하므로 관련 레지스트리 항목은 아래 영구 등록에 따라 업데이트되었다([BCP90] 참조).

Header Field Name	Protocol	Status	Reference
Accept-Ranges	http	standard	<a href="#">Section 2.3</a>
Content-Range	http	standard	<a href="#">Section 4.2</a>
If-Range	http	standard	<a href="#">Section 3.2</a>
Range	http	standard	<a href="#">Section 3.1</a>

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

#### 5.4. Internet Media Type Registration

IANA maintains the registry of Internet media types [BCP13] at <http://www.iana.org/assignments/media-types/>.

IANA는 인터넷 미디어 타입[BCP13]의 Registry를 <http://www.iana.org/assignments/media-types/>에 유지한다.

This document serves as the specification for the Internet media type "multipart/byteranges". The following has been registered with IANA.

이 문서는 인터넷 미디어 타입 "multipart/byteranges"의 명세 역할을 한다. 다음은 IANA에 등록되었다.

#### 5.4.1. Internet Media Type multipart/byteranges

Type name: multipart

Subtype name: byteranges

Required parameters: boundary

Optional parameters: N/A

Encoding considerations: only "7bit", "8bit", or "binary" are permitted

Security considerations: see [Section 6](#)

Interoperability considerations: N/A

Published specification: This specification (see [Appendix A](#)).

Applications that use this media type: HTTP components supporting multiple ranges in a single request.

Fragment identifier considerations: N/A

Additional information:

Deprecated alias names for this type: N/A

Magic number(s): N/A

File extension(s): N/A

Macintosh file type code(s): N/A

Person and email address to contact for further information: See Authors' Addresses section.

Intended usage: COMMON

Restrictions on usage: N/A

Author: See Authors' Addresses section.

Change controller: IESG

## 6. Security Considerations

This section is meant to inform developers, information providers, and users of known security concerns specific to the HTTP range request mechanisms. More general security considerations are addressed in HTTP messaging [[RFC7230](#)] and semantics [[RFC7231](#)].

이 섹션은 HTTP 범위 요청 메커니즘에 특정한 알려진 보안 문제를 개발자, 정보 제공자 및 사용자에게 알리기 위한 것이다. 보다 일반적인 보안 고려사항은 HTTP 메시징 [RFC7230] 및 의미론[RFC7231]에서 다루어진다.

### 6.1. Denial-of-Service Attacks Using Range

Unconstrained multiple range requests are susceptible to denial-of-service attacks because the effort required to request many overlapping ranges of the same data is tiny compared to the time, memory, and bandwidth consumed by attempting to serve the requested data in many parts. Servers ought to ignore, coalesce, or reject egregious range requests, such as requests for more than two overlapping ranges or for many small ranges in a single set, particularly when the ranges are requested out of order for no apparent reason. Multipart range requests are not designed to support random access.

제한되지 않은 다중 범위 요청은 많은 부분에서 요청된 데이터를 서비스하려고 시도함으로써 소비되는 시간, 메모리 및 대역폭에 비해 동일한 데이터의 중복 범위를 많이 요청하는 데 필요한 노력이 작기 때문에 denial-of-service 공격에 취약하다. 서버는 두 개 이상의 겹치는 범위 또는 단일 집합의 많은 작은 범위에 대한 요청과 같은 터무니없는 범위 요청은 무시, 통합 또



는 거부해야 하며, 특히 뚜렷한 이유 없이 범위가 요청될 경우 더욱 그러해야 한다. 멀티파트 범위 요청은 무작위 접근을 지원하도록 설계되지 않았다.

## 7. Acknowledgments

See [Section 10 of \[RFC7230\]](#).

## 8. References

### 8.1. Normative References

[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.

[RFC7230] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", [RFC 7230](#), June 2014.

[RFC7231] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", [RFC 7231](#), June 2014.

[RFC7232] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests", [RFC 7232](#), June 2014.

[RFC7234] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Caching", [RFC 7234](#), June 2014.

## 8.2. Informative References

- [BCP13] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", [BCP 13](#), [RFC 6838](#), January 2013.
- [BCP90] Klyne, G., Nottingham, M., and J. Mogul, "Registration Procedures for Message Header Fields", [BCP 90](#), [RFC 3864](#), September 2004.
- [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.
- [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

When a 206 (Partial Content) response message includes the content of multiple ranges, they are transmitted as body parts in a multipart message body ([\[RFC2046\], Section 5.1](#)) with the media type of "multipart/byteranges".

206 (Partial Content) 응답 메시지가 복수의 범위의 내용을 포함할 때, 그것들은 "multipart/byteranges"의 미디어 타입과 함께 멀티파트 메시지 본문([\[RFC2046\], Section 5.1](#))에서 본문 부분으로 전송된다.

The multipart/byteranges media type includes one or more body 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.

multipart/byteranges 미디어 타입은 하나 이상의 본문 부분을 포함하며, 각각 자신의 Content-Type 및 Content-Range 필드가 있다. 필요한 경계 매개변수는 각 본문 부분을 분리하는 데 사용되는 경계 문자열을 지정한다.

Implementation Notes:

구현 참조:

1. Additional CRLFs might precede the first boundary string in the body.  
1. 추가 CRLF는 신체의 첫 번째 경계 문자열 앞에 있을 수 있다.
2. Although [\[RFC2046\]](#) permits the boundary string to be quoted, some existing implementations handle a quoted boundary string incorrectly.  
2. [RFC2046]에서는 경계 문자열을 따옴표로 묶는 것을 허용하지만, 일부 기존 구현에서는 인용된 경계 문자열을 잘못 취급한다.
3. A number of clients and servers were coded to an early draft of the byteranges specification that used a media type of multipart/x-byteranges, which is almost (but not quite) compatible with this type.  
3. 다수의 클라이언트와 서버가 이 타입과 거의(그러나 완전하지 않은) 호환되는 multipart/x-byteranges의 미디어 타입을 사용하는 byteranges 명세의 초기 초안에 코드화되었다.

Despite the name, the "multipart/byteranges" media type is not limited to byte ranges. The following example uses an "exampleunit" range unit:

명칭에도 불구하고, "multipart/byteranges" 미디어 타입은 바이트 범위에 국한되지 않는다. 다음 예시에서 "exampleunit" 범위 단위를 사용한다.

```
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-Length: 2331785
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--
```

## Appendix B. Changes from [RFC 2616](#)

Servers are given more leeway in how they respond to a range request, in order to mitigate abuse by malicious (or just greedy) clients. ([Section 3.1](#))

서버는 악의적인(또는 단지 탐욕스러운) 클라이언트의 남용을 완화하기 위해 범위 요청에 응답하는 방법에 더 많은 여유가 주어진다. (Section 3.1)

A weak validator cannot be used in a 206 response. ([Section 4.1](#))

206 응답에는 약한 검증자를 사용할 수 없다. (Section 4.1)

The Content-Range header field only has meaning when the status code explicitly defines its use. ([Section 4.2](#))

Content-Range 헤더 필드는 상태 코드가 Content-Range 사용을 명시적으로 정의할 때만 의미가 있다. (Section 4.2)

This specification introduces a Range Unit Registry. ([Section 5.1](#))

이 명세는 Range Unit 레지스트리를 도입한다. (Section 5.1)

multipart/byteranges can consist of a single part. (Appendix A)

multipart/byteranges는 단일 부분으로 구성될 수 있다.(Appendix A)

## Appendix C. Imported ABNF

The following core rules are included by reference, as defined in [Appendix B.1 of \[RFC5234\]](#): 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), and VCHAR (any visible US-ASCII character).

Note that all rules derived from token are to be compared case-insensitively, like range-unit and acceptable-ranges.

The rules below are defined in [\[RFC7230\]](#):

OWS = <OWS, see [\[RFC7230\], Section 3.2.3](#)>

token = <token, see [\[RFC7230\], Section 3.2.6](#)>

The rules below are defined in other parts:

HTTP-date = <HTTP-date, see [\[RFC7231\], Section 7.1.1.1](#)>

entity-tag = <entity-tag, see [\[RFC7232\], Section 2.3](#)>

## Appendix D. Collected ABNF

In the collected ABNF below, list rules are expanded as per [Section 1.2 of \[RFC7230\]](#).

Accept-Ranges = acceptable-ranges

Content-Range = byte-content-range / other-content-range

HTTP-date = <HTTP-date, see [\[RFC7231\], Section 7.1.1.1](#)>

If-Range = entity-tag / HTTP-date

OWS = <OWS, see [\[RFC7230\], Section 3.2.3](#)>

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

entity-tag = <entity-tag, see [\[RFC7232\], Section 2.3](#)>

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

token = <token, see [\[RFC7230\], Section 3.2.6](#)>

unsatisfied-range = "\*" complete-length