HTTP   M. Nottingham
Internet-Draft   March 18, 2020
Intended status: Informational
Expires: September 19, 2020

Advisory Content-Length for HTTP
draft-nottingham-bikeshed-length-00

Abstract

The HTTP Content-Length header field is overloaded with (at least) two duties: message delimitation in HTTP/1, and metadata about the length of an incoming request body to the software handling it.

This causes confusion, and sometimes problems. This document proposes a new header to untangle these semantics (at least partially).

Note to Readers

RFC EDITOR: please remove this section before publication

The issues list for this draft can be found at https://github.com/mnot/I-D/labels/bikeshed-length [1].

The most recent (often, unpublished) draft is at https://mnot.github.io/I-D/bikeshed-length/ [2].

Recent changes are listed at https://github.com/mnot/I-D/commits/gh-pages/bikeshed-length [3].

See also the draft's current status in the IETF datatracker, at https://datatracker.ietf.org/doc/draft-nottingham-bikeshed-length/ [4].

Status of This Memo

Table of Contents

1. Introduction

The HTTP Content-Length header field ([RFC7230]) is overloaded with (at least) two duties: message delimitation in HTTP/1, and metadata about the length of an incoming request body to the software handling it.

Message delimitation is a core feature of the protocol; it allows more than one message to be sent in a given direction on a connection. It is also security-critical; if it is under attacker control, it's possible to confuse a recipient about how requests and responses are associated in HTTP/1.1 (as "smuggling" attacks).

As such, it has been treated progressively more strictly in HTTP specifications. HTTP/1.1 introduced chunked transfer encoding, and forbade sending Content-Length when it is in use. From [RFC2616]:

> Messages MUST NOT include both a Content-Length header field and a non-identity transfer-coding. If the message does include a non-identity transfer-coding, the Content-Length MUST be ignored.
>
> If a message is received with both a Transfer-Encoding header field and a Content-Length header field, the latter MUST be ignored.

[RFC7230] strengthened that to:

> A sender MUST NOT send a Content-Length header field in any message that contains a Transfer-Encoding header field.
>
> If a message is received with both a Transfer-Encoding and a Content-Length header field, the Transfer-Encoding overrides the Content-Length. Such a message might indicate an attempt to perform request smuggling (Section 9.5) or response splitting (Section 9.4) and ought to be handled as an error. A sender MUST remove the received Content-Length field prior to forwarding such a message downstream.

HTTP/2 ([RFC7540]) does not use Content-Length for message delimitation, but still requires it to match the number of bytes that its framing mechanism sends:

> A request or response that includes a payload body can include a content-length header field. A request or response is also malformed if the value of a content-length header field does not equal the sum of the DATA frame payload lengths that form the body.

It further requires such malformed responses to generate a "hard" error, so that a downstream recipient that implements HTTP/1 can't be attacked:

> Intermediaries that process HTTP requests or responses (i.e., any intermediary not acting as a tunnel) MUST NOT forward a malformed request or response. Malformed requests or responses that are detected MUST be treated as a stream error (Section 5.4.2) of type PROTOCOL_ERROR.
>
> For malformed requests, a server MAY send an HTTP response prior to closing or resetting the stream. Clients MUST NOT accept a malformed response. Note that these requirements are intended to

protect against several types of common attacks against HTTP; they are deliberately strict because being permissive can expose implementations to these vulnerabilities.

The currently proposed HTTP/3 specification [I-D.ietf-quic-http] has language similar to that in HTTP/2.

Unfortunately, this makes _other_ uses of Content-Length more difficult to implement.

In particular, many servers will reject a request without an explicit Content-Length using 411 (Length Required), because they want to know how many bytes are being sent before deciding to devote resources to serving the request. However, depending on the protocol version(s) between the user agent and the origin server, a Content-Length header might not make it all the way, or the request might be rejected.

Likewise, some applications would like to use Content-Length to indicate progress of a large download, but its successful traversal cannot be relied upon.

While it's questionable whether all of the requirements above regarding Content-Length are honoured by implementations uniformly, there is enough diversity in implementation (particularly on the server side and in intermediaries) to make deployment of these uses daunting.

Therefore, this specification proposes a new HTTP header field to carry _advisory_ content length information. It is intended only for these uses, and _not_ message delimitation.

1.1. Notational Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. The Bikeshed-Length HTTP Header Field

NOTE: The final name of this header field will be selected using a to-be-defined process. Warm up your paintbrushes...

The Bikeshed-Length HTTP header field is a HTTP Structured Field [I-D.ietf-httpbis-header-structure] that conveys an advisory content length for the message body:

```
Bikeshed-Length = sh-item
```

Its value MUST be an Integer (Section x.x of [I-D.ietf-httpbis-header-structure]), indicating a decimal number of octets for a potential payload body.

Note that it is specifically a header field; it is not allowed to occur in trailer sections, and SHOULD be ignored if encountered there.

2.1. Example

A resource might allow requests with bodies up to a given size. If an incoming request omits both Content-Length and Bikeshed-Length, they can respond with 411 (Length Required). If either request header field is present, and the value given is not acceptable, they can respond with 413 (Payload Too Large). If Bikeshed-Length is used and deemed to be acceptable, the resource still ought to monitor the number of incoming bytes to assure that they do not exceed the anticipated value.

3. IANA Considerations

TBD

4. Security Considerations

The Value of Bikeshed-Length is advisory only; software that uses it will need to monitor the actual number of octets received to assure that it is not exceeded, and take appropriate action if it is.

5. References

5.1. Normative References

[I-D.ietf-httpbis-header-structure]
Nottingham, M. and P. Kamp, "Structured Field Values for HTTP", draft-ietf-httpbis-header-structure-17 (work in progress), March 2020.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC7230] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, DOI 10.17487/RFC7230, June 2014, <https://www.rfc-editor.org/info/rfc7230>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

5.2. Informative References

[I-D.ietf-quic-http] Bishop, M., "Hypertext Transfer Protocol Version 3 (HTTP/3)", draft-ietf-quic-http-27 (work in progress), February 2020.

[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, DOI 10.17487/RFC2616, June 1999, <https://www.rfc-editor.org/info/rfc2616>.

[RFC7540] Belshe, M., Peon, R., and M. Thomson, Ed., "Hypertext Transfer Protocol Version 2 (HTTP/2)", RFC 7540, DOI 10.17487/RFC7540, May 2015, <https://www.rfc-editor.org/info/rfc7540>.

5.3. URIs

[1] https://github.com/mnot/I-D/labels/bikeshed-length

[2] https://mnot.github.io/I-D/bikeshed-length/

[3] https://github.com/mnot/I-D/commits/gh-pages/bikeshed-length

[4] https://datatracker.ietf.org/doc/draft-nottingham-bikeshed-length/

Author's Address

Mark Nottingham

Email: mnot@mnot.net
URI: https://www.mnot.net/