Internet Engineering Task Force (IETF)
Request for Comments: 7694
Category: Standards Track
ISSN: 2070-1721

J. Reschke
greenbytes
November 2015

Hypertext Transfer Protocol (HTTP) Client-Initiated Content-Encoding

Abstract

In HTTP, content codings allow for payload encodings such as for compression or integrity checks. In particular, the "gzip" content coding is widely used for payload data sent in response messages.

Content codings can be used in request messages as well; however, discoverability is not on par with response messages. This document extends the HTTP "Accept-Encoding" header field for use in responses, to indicate the content codings that are supported in requests.

Status of This Memo

This is an Internet Standards Track document.

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[1].

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/rfc7694[2].

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info[3]) 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.

[1] https://tools.ietf.org/html/rfc5741#section-2
[2] http://www.rfc-editor.org/info/rfc7694
[3] http://trustee.ietf.org/license-info

Table of Contents

1 Introduction....3

2 Notational Conventions....4

3 Using the 'Accept-Encoding' Header Field in Responses....5

4 Example....6

5 Deployment Considerations....7

6 Security Considerations....8

7 IANA Considerations....9

7.1 Header Field Registry....9

7.2 Status Code Registry....9

8 References....10

8.1 Normative References....10

8.2 Informative References....10

Author's Address....12

1. Introduction

In HTTP, content codings allow for payload encodings such as for compression or integrity checks ([RFC7231], Section 3.1.2). In particular, the "gzip" content coding ([RFC7230], Section 4.2) is widely used for payload data sent in response messages.

Content codings can be used in request messages as well; however, discoverability is not on par with response messages. This document extends the HTTP "Accept-Encoding" header field ([RFC7231], Section 5.3.4) for use in responses, to indicate the content codings that are supported in requests. It furthermore updates the definition of status code 415 (Unsupported Media Type) ([RFC7231], Section 6.5.13), recommending that the "Accept-Encoding" header field be included when appropriate.

2. Notational Conventions

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].

This document reuses terminology defined in the base HTTP specifications, namely Section 2 of [RFC7230] and Section 3.1.2 of [RFC7231].

3. Using the 'Accept-Encoding' Header Field in Responses

Section 5.3.4 of [RFC7231] defines "Accept-Encoding" as a request header field only.

This specification expands that definition to allow "Accept-Encoding" as a response header field as well. When present in a response, it indicates what content codings the resource was willing to accept in the associated request. A field value that only contains "identity" implies that no content codings were supported.

Note that this information is specific to the associated request; the set of supported encodings might be different for other resources on the same server and could change over time or depend on other aspects of the request (such as the request method).

Section 6.5.13 of [RFC7231] defines status code 415 (Unsupported Media Type) to apply to problems related to both media types and content codings.

Servers that fail a request due to an unsupported content coding ought to respond with a 415 status and ought to include an "Accept-Encoding" header field in that response, allowing clients to distinguish between issues related to content codings and media types. In order to avoid confusion with issues related to media types, servers that fail a request with a 415 status for reasons unrelated to content codings MUST NOT include the "Accept-Encoding" header field.

It is expected that the most common use of "Accept-Encoding" in responses will have the 415 (Unsupported Media Type) status code, in response to optimistic use of a content coding by clients. However, the header field can also be used to indicate to clients that content codings are supported, to optimize future interactions. For example, a resource might include it in a 2xx response when the request payload was big enough to justify use of a compression coding but the client failed do so.

4. Example

A client submits a POST request using the "compress" content coding ([RFC7231], Section 3.1.2.1):

```
POST /edit/ HTTP/1.1
Host: example.org
Content-Type: application/atom+xml;type=entry
Content-Encoding: compress

...compressed payload...
```

The server rejects the request because it only allows the "gzip" content coding:

```
HTTP/1.1 415 Unsupported Media Type
Date: Fri, 09 May 2014 11:43:53 GMT
Accept-Encoding: gzip
Content-Length: 68
Content-Type: text/plain

This resource only supports the "gzip" content coding in requests.
```

At this point, the client can retry the request with the supported "gzip" content coding.

Alternatively, a server that does not support any content codings in requests could answer with:

```
HTTP/1.1 415 Unsupported Media Type
Date: Fri, 09 May 2014 11:43:53 GMT
Accept-Encoding: identity
Content-Length: 61
Content-Type: text/plain

This resource does not support content codings in requests.
```

5. Deployment Considerations

Servers that do not support content codings in requests already are required to fail a request that uses a content coding. Section 6.5.13 of [RFC7231] defines the status code 415 (Unsupported Media Type) for this purpose, so the only change needed is to include the "Accept-Encoding" header field with the value "identity" in that response.

Servers that do support some content codings are required to fail requests with unsupported content codings as well. To be compliant with this specification, servers will need to use the status code 415 (Unsupported Media Type) to signal the problem and will have to include an "Accept-Encoding" header field that enumerates the content codings that are supported. As the set of supported content codings is usually static and small, adding the header field ought to be trivial.

6. Security Considerations

This specification only adds discovery of supported content codings and diagnostics for requests failing due to unsupported content codings. As such, it doesn't introduce any new security considerations over those already present in HTTP/1.1 (Section 9 of [RFC7231]) and HTTP/2 (Section 10 of [RFC7540]).

However, the point of better discoverability and diagnostics is to make it easier to use content codings in requests. This might lead to increased usage of compression codings such as gzip (Section 4.2 of [RFC7230]), which, when used over a secure channel, can enable side-channel attacks such as BREACH (see Section 10.6 of [RFC7540] and [BREACH]). At the time of publication, it was unclear how BREACH-like attacks can be applied to compression in HTTP requests.

7. IANA Considerations

7.1. Header Field Registry

HTTP header fields are registered within the "Message Headers" registry located at <http://www.iana.org/assignments/message-headers>, as defined by [BCP90].

This document updates the definition of the "Accept-Encoding" header field. The "Permanent Message Header Field Names" registry has been updated as follows:

Header Field Name	Protocol	Status	Reference
Accept-Encoding	http	standard	Section 5.3.4 of [RFC7231] and Section 3 of this document

7.2. Status Code Registry

HTTP status codes are registered within the "HTTP Status Codes" registry located at <http://www.iana.org/assignments/http-status-codes>.

This document updates the definition of the status code 415 (Unsupported Media Type). The "HTTP Status Codes" registry has been updated as follows:

Value	Description	Reference
415	Unsupported Media Type	Section 6.5.13 of [RFC7231] and Section 3 of this document

8. References

8.1. Normative References

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

[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, <http://www.rfc-editor.org/info/rfc>.

[RFC7231] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", RFC 7231, DOI 10.17487/RFC7231, June 2014, <http://www.rfc-editor.org/info/rfc>.

8.2. Informative References

[BCP90] Klyne, G., Nottingham, M., and J. Mogul, "Registration Procedures for Message Header Fields", BCP 90, RFC 3864, September 2004, <http://www.rfc-editor.org/info/bcp>.

[BREACH] Gluck, Y., Harris, N., and A. Prado, "BREACH: Reviving the CRIME Attack", July 2013, <http://breachattack.com/resources/BREACH%20-%20SSL,%20gone%20in%2030%20seconds.pdf>.

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

Acknowledgements

Thanks go to the Hypertext Transfer Protocol Working Group participants, namely Amos Jeffries, Ben Campbell, Mark Nottingham, Pete Resnick, Stephen Farrell, and Ted Hardie.

Author's Address

Julian F. Reschke
greenbytes GmbH
Hafenweg 16
Muenster, NW 48155
Germany
Email: julian.reschke@greenbytes.de
URI: http://greenbytes.de/tech/webdav/