Digest Headers

(was: Resource Digests, was: RFC 3230)

HTTPWG Interim 2021-02

draft-ietf-httpbis-digest-headers

[last interim slides] [latest editor copy]

Since October 2020 Interim

Mainly editorial changes, sitting in the editor's copy.

Thanks for the reviews and feedback.

See

https://tools.ietf.org/rfcdiff?url1=https://tools.ietf.org/id/draft-ietf-httpbis-digest-headers-04.txt&url2=https://httpwg.org/http-extensions/draft-ietf-httpbis-digest-headers.txt

Open Issues Needing Input

- Digest in requests
- Obsoleted algorithms

Digest vs Content-MD5 messages

Header	Spec	Status	Notes
Digest	RFC3230	Standard	Computed on "instance", aka (selected) representation data in RFC7231 terms. HTTP-semantics aware.
Content-MD5	RFC1864 RFC2616	Deprecated	**Computed on payload data**. Deprecated by RFC7231 Appendix B because **inconsistently implemented with respect to partial responses**.

What does Digest in requests mean?

Several issues related to digest and requests

#970, #1005, #1357, #1366

Unsticking these problems gets us much closer to being done.

Example use case: Resumable upload

A client uploads different ranges (chunks) of a large file. This supports a resumable upload model.

It wants to use a digest to help the upload process validate the integrity of [each chunk, the reassembled file, something else …]

Partial requests and http-semantics

Recently documented in http-semantics § 14.4 and 14.5

a. Servers MUST ignore Content-range in requests with a method that does not define it. "*No request method in this specification is defined to support Content-Range*".
b. Partial PUT (with a Content-range) is supported by some. "*though such support is inconsistent and depends on private agreements with user agents*".

Digest in Requests: schools of thought

Possible behaviors	Pros	Cons
Always computed on payload data	* Easier to implement for servers and intermediaries on requests	* Resurrect the Content-MD5 behavior. * Asymmetric between request and response. * Response handling still requires ability to distinguish between complete and partial representation
Computed on the representation data	* Maintains RFC3230 intent * Coherent definition for request and response. * Behavior can adapt to requests conveying partial representations * Still computed on payload data when conveying complete representations	* Intermediaries implementing Digest should be capable to distinguish when a request conveys a partial representation on both Requests and Responses

Possible path forward

1. Agree digest applies to request representation data.
2. Appreciate that “Partial PUT” is not commonly solved.
    a. Use of Digests in requests today is typically done on the full representation data, which is equivalent to payload data.
3. Digest spec makes progress.
4. (if needed) Some future Method can standardize partial requests.
    a. Activity should also consider if an integrity check of the payload data is also useful. E.g. new header required to carry hash of payload data **on both requests and responses**.

Mid-deck break

Any strong opinions not yet shared?

Dealing with old algorithms (1) #1377

Current

Algorithm	Status
sha-256	standard
sha-512	standard
md5	deprecated
sha	deprecated
unixsum	standard
unixcksum	standard
crc32c	standard
adler32	obsoleted
contentMD5	obsoleted*

- standard (fine to use)
- deprecated (MUST NOT use)
- obsoleted (SHOULD NOT use)
- obsoleted* (MUST NOT use)

Dealing with old algorithms (2) #1377

Current

Algorithm	Status
sha-256	standard
sha-512	standard
md5	deprecated
sha	deprecated
unixsum	standard
unixcksum	standard
crc32c	standard
adler32	obsoleted
contentMD5	obsoleted*

Status is confusing

- standard (fine to use)
- deprecated (MUST NOT use)
- obsoleted (SHOULD NOT use)
- obsoleted* (MUST NOT use)

Dealing with old algorithms (3) #1377

Proposed

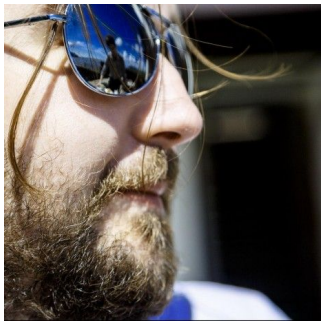
Algorithm	Status
sha-256	standard
sha-512	standard
md5	obsoleted
sha	obsoleted
unixsum	obsoleted
unixcksum	obsoleted
crc32c	obsoleted
adler32	obsoleted
contentMD5	obsoleted

- standard (fine to use)
- obsoleted (MUST NOT use)

Thanks!

Roberto Polli - robipolli@gmail.com

Lucas Pardue - lucaspardue.24.7@gmail.com



Backlog

id- prefix for digest-algorithms: should we strip id-sha-256? #885