

# Digest Headers

(was: Resource Digests, was: RFC 3230)

IETF 106 Singapore

draft-ietf-httpbis-digest-headers

[\[see IETF105 slides\]](#) [\[see the specifications\]](#)

# Digest HTTP Header Field summary

Request:

```
GET /items/123
```

Response:

```
HTTP/1.1 200 Ok
```

```
Content-Type: application/json
```

```
Content-Encoding: identity
```

```
Digest: sha-256=X48E9q0okqrvdts8n0JRJN30WUoyWxBf7kbu9DBPE=
```

```
{"hello": "world"}
```

**digest-algorithm**



**encoded digest output**



# Who is using Digest?

- [MICE content-coding](#) (draft-thomson-http-mice)
- Signature specs: http-signatures, [signed-exchanges](#) (draft-yasskin-http-origin-signed-responses)
- Banking APIs via http-signatures

# Changes in 01

- Editorial sweep
  1. Clarify state-changing methods
  2. Reboot digest-algorithm IANA table
  3. Relationship with Subresource Integrity (SRI)

# Change 1: Clarify state-changing methods

Issue [#853](#)

POST and PATCH requests convey actions, not partial representations. Digest is then computed:

- in requests, on the representation-data of those actions.
- in responses: on the selected representation of the referenced resource. This may be the enclosed OR the selected representation (eg. in case of 204 No Content).

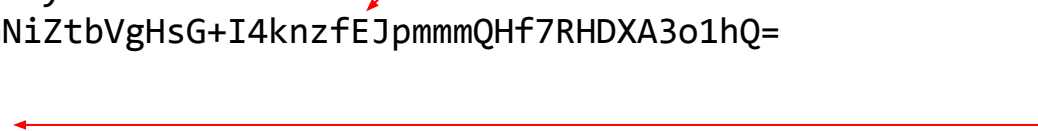

# Change 1: POST example

Request:

```
POST /books/123 HTTP/1.1
Content-Type: application/json
Accept: application/json
Accept-Encoding: identity
Digest: sha-256=bWopGGNiZtbVgHsG+I4knzfEJpmmmQHf7RHDXA3o1hQ=
```

```
{"title": "New Title"}
```

Request digest applies to enclosed representation




Response:

```
HTTP/1.1 201 Created
Content-Type: application/json
Digest: id-sha-256=0o/WKwSfnmIoSlop2LV/ISaBDth05IeW27zzNMUh518=
Location: /books/123
```

```
{"status": "created", "id": "123", "ts": 1569327729, "instance": "/books/123"}
```

Response digest applies to enclosed representation



# Change 1: PATCH example

Request:

```
PATCH /books/123 HTTP/1.1
Content-Type: application/merge-patch+json
Accept: application/json
Accept-Encoding: identity
Digest: sha-256=bWopGGNiZtbVgHsG+I4knzfEJpmmmQHf7RHDXA3o1hQ=
```

JSON patch (RFC 7396)

Request digest applies to  
patch document

```
{"title": "New Title"}
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Digest: id-sha-256=BZlF2v0IzjuxN01RQ97EUXriaNNLhtI8Chx8Eq+XYSc=
```

Response digest applies to complete  
representation of patched document

```
{"id": "123", "title": "New Title"}
```

# Change 1: PATCH example with 204

Request:

```
PATCH /books/123 HTTP/1.1
Content-Type: application/merge-patch+json
Accept: application/json
Accept-Encoding: identity
Digest: sha-256=bWopGGNiZtbVgHsG+I4knzfEJpmmmQHf7RHDXA3o1hQ=
```

JSON patch (RFC 7396)

Request digest applies to  
patch document

```
{"title": "New Title"}
```

Response:

```
HTTP/1.1 204 No Content
Content-Type: application/json
Digest: id-sha-256=BZlF2v0IzjuxN01RQ97EUXriaNNLhtI8Chx8Eq+XYSc=
```

Response digest applies to complete  
representation of patched document but no  
payload provided



# Change 1: Open Issue [#970](#) - Is POST behavior extensible to all payload bodies?

Julian - *“I just don't think that it would be a good idea to vary the semantics based on the request method.”*

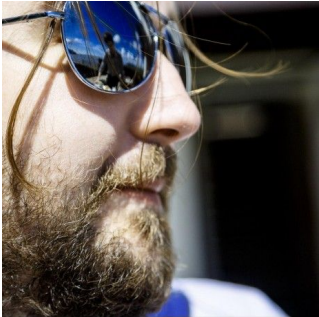
We can address this with some rewording but should we? E.g

Does a present or future method convey a partial representation, and if so the digest should always be computed on the complete representation.

# Thanks!

Roberto Polli - [robipolli@gmail.com](mailto:robipolli@gmail.com)

Lucas Pardue - [lucaspardue.24.7@gmail.com](mailto:lucaspardue.24.7@gmail.com)



© rjccartoons | Dreamstime.com



# Change 2: Reboot digest-algorithm IANA table

- New "status" field to mark deprecated/obsoleted algorithms
- Deprecate MD5 as a weak crypto algorithm (issue [#867](#))
- Obsolete SHA and ADLER32 as there are better replacements (issue [#828](#))
- Simplified citation of SHA (issue [#832](#))

# Open Issues Needing Input

- [#936/#937](#) - Cache and Digest
- [#851](#) - detail more the use with HTTP signatures
- [#852](#) - add a threat model?
- [#849](#) - digest of an empty representation
- [#850](#) - digest-algorithm “parameter” spec gap
- [#970](#) - ~~Is POST behavior extensible to all payload bodies?~~  
(already mentioned)

# [#936](#)/[#937](#) - Cache, Digest and cache-validators

RFC 3230 states the following:

The instance is specified by the Request-URI and **any cache-validator** contained in the message.

we translated it in to RFC 723x terms:

The resource is specified by the effective request URI and **any `validator`** contained in the message.

But how **do** validators specify a resource? Is "specify" the correct term?

# #851 - using Digest in signatures

- Digest main use case is with HTTP signatures
- 01 provides minimal guidance:
  - use transport integrity, sign data and metadata, avoid broken algorithms.
- Are there compelling reasons to expand on this?
  - Especially guidance related to representation-metadata e.g. Content-Length

## #852 - add a threat model?

- Is a threat model useful?
- Should we document it in this I-D?
- We have some candidate text already on the issue so next steps might be:
  - a. Close, not needed
  - b. Move to a PR
  - c. Consider a broader threat modelling (see relationship to HTTP signatures issues)

## #849 - digest of an empty representation

More confusing than it sounds, would examples help?

One case: an empty representation may have a non-empty body due to content-encoding, affecting Digest value.

```
>>> sha256(compress(b'')).hexdigest()  
'7a53d5f4237c606ddaba52a2d4a3e40200eea48f5992172c6751209decae8d5a'
```

```
>>> sha256(b'').hexdigest()  
'e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855'
```



# #850 - digest-algorithm “parameter” spec gap

RFC3230 states the following and we import it verbatim:

For some algorithms, one or more parameters may be supplied.

```
digest-algorithm = token
```

The BNF for "parameter" is as is used in RFC 2616 [4]. All digest-algorithm values are case-insensitive.

Problems:

No example of parameter, anywhere.

Reference to BNF needs updating