Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 21 June 2021

R. Polli
Digital Transformation Department, Italian Government

18 December 2020

# The "id-" prefix for Digest Algorithms
## draft-polli-id-digest-algorithms-01

## Abstract

This document defines the "id-" prefix for digest-algorithms used in the Digest HTTP field. This prefix explicits that the value of the digest-algorithm is independent from Content-Encoding.

## Note to Readers

_RFC EDITOR: please remove this section before publication_

Discussion of this draft takes place on the HTTP working group mailing list (ietf-http-wg@w3.org), which is archived at https://lists.w3.org/Archives/Public/ietf-http-wg/ (https://lists.w3.org/Archives/Public/ietf-http-wg/).

The source code and issues list for this draft can be found at https://github.com/ioggstream/draft-polli-Retry-Scope (https://github.com/ioggstream/draft-polli-Retry-Scope).

## Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 21 June 2021.

Table of Contents



## 1. Introduction

The [DIGEST] defines a way to convey a checksum of a representation-data as specified in [SEMANTICS].

As the representation data depends on the value of "Content-Encoding", it is useful to convey the checksum value of a representation without any content-coding applied.

This proposal introduces the "id-" prefix to specify that the provided digest-algorithm value is computed on the representation-

data without any content-coding applied.





### 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. These words may also appear in this document in lower case as plain English words, absent their normative meanings.

This document uses the Augmented BNF defined in [RFC5234] and updated by [RFC7405].

The definitions "representation", "selected representation", "representation data", "representation metadata", and "payload body" in this document are to be interpreted as described in [SEMANTICS].

The definitions "digest-algorithm" and "representation-data-digest" in this document are to be interpreted as described in [DIGEST].

## 2. The "id-" prefix for digest-algorithms

A digest-algorithm to be registered within the HTTP Digest Algorithm Values (https://www.iana.org/assignments/http-dig-alg/http-dig-alg.xhtml) MUST NOT start with the string "id-".

The following two examples show two digest-algorithm names that cannot be registered

```
id-crc32c
id-adler32
```

For every digest-algorithm registered in the HTTP Digest Algorithm Values (https://www.iana.org/assignments/http-dig-alg/http-dig-alg.xhtml) the associate "id-" digest-algorithm has the following properties:

* the checksum is computed on the representation-data of the resource when no content coding is applied;

* the checksum is computed according to the original digest-algorithm Description field, and uses the same encoding of the original digest-algorithm.

This definition is compatible, and thus extends, the definition of the "id-sha-256" and "id-sha-512" digest-algorithms contained in Section X of [DIGEST].





## 3. Security Considerations

### 3.1. Disclosure of encrypted content

Like the "id-sha-256" digest-algoritm defined in [DIGEST] if the content-coding provides encryption features, sending the checksum of unencoded representation can disclose information.

## 4. IANA Considerations

### 4.1. TBD how to reserve "id-" prefix?

## 5. Examples

### 5.1. The id-crc32c digest-algorithm

The following request conveys a brotli encoded json object

{"hello": "world"}

The "Digest" computed using the "crc32c" digest-algorithm present in HTTP Digest Algorithm Values (https://www.iana.org/assignments/http-dig-alg/http-dig-alg.xhtml) is content-coding aware, while its associated "id-" digest-algorithm is not "id-crc32c"

```
POST /data HTTP/1.1
Content-Type: application/json
Content-Encoding: br
Digest: id-crc32c=43794720, crc32c=DB329237
```

CwGAZG9nAw==

## 6. Normative References


[DIGEST] Polli, R. and L. Pardue, "Digest Headers", Work in Progress, Internet-Draft, draft-ietf-httpbis-digest-headers-04, 17 October 2020, <http://www.ietf.org/internet-drafts/draft-ietf-httpbis-digest-headers-04.txt>.

[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>.

[RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <https://www.rfc-editor.org/info/rfc5234>.







[RFC7405] Kyzivat, P., "Case-Sensitive String Support in ABNF", RFC 7405, DOI 10.17487/RFC7405, December 2014, <https://www.rfc-editor.org/info/rfc7405>.

[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>.

[SEMANTICS] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", RFC 7231, DOI 10.17487/RFC7231, June 2014, <https://www.rfc-editor.org/info/rfc7231>.


## Appendix A. Acknowledgements


This specification was born from a thread created by James Manger and the subsequent discussion here https://github.com/httpwg/http-extensions/issues/885.


FAQ

Q: Question 1  Answer 1

## Code Samples

_RFC Editor: Please remove this section before publication._

How can I generate and validate the "Digest" values shown in the examples throughout this document?

The following python3 code can be used to generate digests for json objects using crc32c algorithm.  Note that these are formatted as base64.  This function could be adapted to other algorithms and should take into account their specific formatting rules.





```
import base64, json, brotli, crc32c

identity = lambda x: x

def digest(item, content_coding=identity, algorithm=crc32c.crc32c):
    json_bytes = json.dumps(item).encode()
    content_encoded = content_coding(json_bytes)
    checksum = algorithm(content_encoded)
    # encode result has uppercase hex
    return hex(checksum)[2:].upper()


item = {"hello": "world"}

print("crc32c digest value for a br-coded representation: ",
```

```
        digest(item, content_coding=brotli.compress)
    )

    print("id-crc32c digest value for a br-coded representation: ",
        digest(item, content_coding=identity)
    )
```

## Change Log

RFC EDITOR PLEASE DELETE THIS SECTION.

## Author's Address

Roberto Polli
Digital Transformation Department, Italian Government
Italy

Email: robipolli@gmail.com