

Workgroup: Network Working Group
Internet-Draft:
draft-bormann-core-media-content-type-
format-02

Published: 18 August 2020

Intended Status: Standards Track

Expires: 19 February 2021

Authors: C. Bormann
Universität Bremen TZI

On Media-Types, Content-Types, and related terminology

Abstract

There is a lot of confusion about media-types, content-types, and related terminology.

This memo is an attempt at clearing it up, so we can use consistent terminology in CoRE and related specifications. It also defines some ABNF that can be used in these specifications.

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 19 February 2021.

Copyright Notice

Copyright (c) 2020 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 (<https://trustee.ietf.org/license-info>) 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.

Table of Contents

- [1. Introduction](#)
- [2. Media-Type](#)
- [3. Content-Type](#)
- [4. Content-Coding](#)
- [5. Content-Format](#)
- [6. Abbreviations](#)
- [7. Discussion](#)
- [8. IANA Considerations](#)
- [9. Security Considerations](#)
- [10. References](#)
 - [10.1. Normative References](#)
 - [10.2. Informative References](#)
- [Acknowledgements](#)
- [Author's Address](#)

1. Introduction

[[RFC1590](#)] introduced media types and their registration. That document took MIME types from [[RFC1521](#)] and gave them a new name. At that time, the term "media type" was often used just for the major type ("text", "audio"), and what we call a media-type now was the combination of a type and a subtype. This lives on in [[RFC6838](#)], which does not even have an ABNF [[RFC5234](#)] production for media type. [[RFC6838](#)]'s predecessor, [[RFC4288](#)], supplied the ABNF shown in ([Figure 1](#)).

```
type-name = reg-name
subtype-name = reg-name

reg-name = 1*127reg-name-chars
reg-name-chars = ALPHA / DIGIT / "!" /
                "#" / "$" / "&" / "." /
                "+" / "-" / "^" / "_"
```

Figure 1: ABNF for type and subtype, from RFC 4288

[[RFC4288](#)] contains the semantically equivalent ABNF in [Figure 2](#), which also provides comments that limit the use of "." and "+".

```

type-name = restricted-name
subtype-name = restricted-name

restricted-name = restricted-name-first *126restricted-name-chars
restricted-name-first = ALPHA / DIGIT
restricted-name-chars = ALPHA / DIGIT / "!" / "#" /
                        "$" / "&" / "-" / "^" / "_"
restricted-name-chars =/ "." ; Characters before first dot always
                        ; specify a facet name
restricted-name-chars =/ "+" ; Characters after last plus always
                        ; specify a structured syntax suffix

```

Figure 2: ABNF for type and subtype, from RFC 6838

2. Media-Type

However, the term "media type" is now generally used for a registered combination of a type-name and a subtype-name. We further disambiguate by calling this a media type name, as, in ABNF:

```
Media-Type-Name = type-name "/" subtype-name
```

For the purposes of this memo, we define:

Media-Type-Name: A combination of a type-name and a subtype-name registered in [[IANA.media-types](#)], conventionally identified by the two names separated by a slash.

(This leaves the term "Media Type" for the actual specification that is registered under the Media-Type-Name.)

3. Content-Type

Media types can have parameters [[RFC6838](#)], some of which are mandatory. In HTTP and many other protocols, these are then used in a "Content-Type" header field. HTTP [[RFC7231](#)] uses:

```

Content-Type = media-type
media-type = type "/" subtype *( OWS ";" OWS parameter )
type       = token
subtype    = token
token      = 1*tchar
tchar      = "!" / "#" / "$" / "%" / "&" / "'" / "*"
            / "+" / "-" / "." / "^" / "_" / "`" / "|" / "~"
            / DIGIT / ALPHA
OWS        = *( SP / HTAB )

```

Figure 3: Content-Type ABNF from RFC 7231

We don't follow this inclusive use established by [\[RFC2616\]](#), parts of which became [\[RFC7231\]](#), namely to use the term media-type for a Media-Type-Name with parameters; note that [\[RFC2616\]](#) was quite confused about this by claiming (Section 3.7):

Media-type values are registered with the Internet Assigned Number Authority (IANA [19]).

This clearly reverts to the understanding of Media-Type-Name we use. We instead define as a separate term:

Content-Type: A Media-Type-Name, optionally associated with parameters (separated from the media type name and from each other by a semicolon).

Removing the legacy HTAB characters now shunned in polite conversation, as well as some other cobwebs, we define the conventional textual representation of a Content-Type as:

```
Content-Type    = Media-Type-Name *( *SP ";" *SP parameter )
parameter      = token "=" ( token / quoted-string )

token          = 1*tchar
tchar          = "!" / "#" / "$" / "%" / "&" / "'" / "*"
               / "+" / "-" / "." / "^" / "_" / "`" / "|" / "~"
               / DIGIT / ALPHA
quoted-string  = %x22 *qdttext %x22
qdttext       = SP / %x21 / %x23-5B / %x5D-7E
```

Note that there is a slight inconsistency between the "token" used here and the "reg-name"/"restricted-name" used above; since media type parameters probably will be defined within the guard rails set by [\[RFC7231\]](#), we need to use HTTP's more comprehensive definition here.

4. Content-Coding

[\[RFC2616\]](#) also introduced the term Content-Coding, a registered name for an encoding transformation that has been or can be applied to a representation:

```
content-coding = token
```

Confusingly, in HTTP the Content-Coding is then given in a header field called "Content-Encoding"; we NEVER use this term (except when we are in error). Instead we define:

Content-Coding: a registered name for an encoding transformation that has been or can be applied to a representation.

Content-Codings are registered in the HTTP Content Coding Registry, a subregistry of [[IANA.http-parameters](#)]. We often use the "identity" Content-Coding, which is the identity transformation, and often fail to identify that Content-Coding by name, instead calling it "no Content-Coding".

5. Content-Format

CoAP [[RFC7252](#)] defines a Content-Format as the combination of a Content-Type and a Content-Coding, identified by a numeric identifier defined by the "CoAP Content-Formats" registry (a subregistry of [[IANA.core-parameters](#)]), but in more confusing words (it did not have the benefit of the present memo).

Content-Format: the combination of a Content-Type and a Content-Coding, identified by a numeric identifier defined by the "CoAP Content-Formats" registry.

Note that there has not been a conventional string representation of just the combination of a Content-Type and a Content-Coding; Content-Formats so far always are identified by their registered Content-Format numbers. However, there are applications where that is useful [[I-D.keranen-core-senml-data-ct](#)], so we define:

```
Content-Format = 1*DIGIT
Content-Format-String = Content-Type ["@" content-coding]
```

This allows the use of Content-Format-Strings such as "application/json@deflate" in place of the less self-describing content-format "11050", or other combinations that do not have a content-format number defined yet.

Content-Format-Strings MUST NOT explicitly use the content-coding value of "identity" (i.e., if an identity content-coding is desired, the entire optional part including the "@" sign is left out).

Note that a quoted string inside a content-type parameter might contain an "@" sign, so the parsing of Content-Format-Strings cannot be done in a too simplistic way.

6. Abbreviations

Media type names are sometimes abbreviated as "mt", and Content-Types as "ct". We propose not to use those abbreviations: Where the long form of the values can be used, the long form "Content-Type" can also be used to name them.

For historical reasons, both [[RFC6690](#)] and [[RFC7252](#)] use the abbreviation "ct" for Content-Format (think first and last character).

For Content-Coding, the abbreviation "cc" can be used.

7. Discussion

The ABNF given here is provisional and needs to be cleaned up: We need to unify the various forms of reg-name, token, etc.

(ABNF just shown for illustration is centered and tagged with "abnf;old" in the XML, while the normative ABNF of this memo is left-aligned and tagged with "abnf".)

We need to discuss case-insensitivity, which is usually rather insensitive.

8. IANA Considerations

While this memo talks a lot about IANA registries, it does not require any action from IANA.

9. Security Considerations

Confusion about terminology may, in the worst case, cause security problems. No other security considerations are known to be raised by the present memo.

10. References

10.1. Normative References

[IANA.core-parameters] IANA, "Constrained RESTful Environments (CoRE) Parameters", , <<http://www.iana.org/assignments/core-parameters>>.

[IANA.http-parameters] IANA, "Hypertext Transfer Protocol (HTTP) Parameters", , <<http://www.iana.org/assignments/http-parameters>>.

[IANA.media-types] IANA, "Media Types", , <<http://www.iana.org/assignments/media-types>>.

10.2. Informative References

[I-D.keranen-core-senml-data-ct]

Keranen, A. and C. Bormann, "SenML Data Value Content-Format Indication", Work in Progress, Internet-Draft, draft-keranen-core-senml-data-ct-02, 8 July 2019,

<<http://www.ietf.org/internet-drafts/draft-keranen-core-senml-data-ct-02.txt>>.

- [RFC1521] Borenstein, N. and N. Freed, "MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies", RFC 1521, DOI 10.17487/RFC1521, September 1993, <<https://www.rfc-editor.org/info/rfc1521>>.
- [RFC1590] Postel, J., "Media Type Registration Procedure", RFC 1590, DOI 10.17487/RFC1590, March 1994, <<https://www.rfc-editor.org/info/rfc1590>>.
- [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>>.
- [RFC4288] Freed, N. and J. Klensin, "Media Type Specifications and Registration Procedures", RFC 4288, DOI 10.17487/RFC4288, December 2005, <<https://www.rfc-editor.org/info/rfc4288>>.
- [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>>.
- [RFC6690] Shelby, Z., "Constrained RESTful Environments (CoRE) Link Format", RFC 6690, DOI 10.17487/RFC6690, August 2012, <<https://www.rfc-editor.org/info/rfc6690>>.
- [RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, DOI 10.17487/RFC6838, January 2013, <<https://www.rfc-editor.org/info/rfc6838>>.
- [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, <<https://www.rfc-editor.org/info/rfc7231>>.
- [RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, DOI 10.17487/RFC7252, June 2014, <<https://www.rfc-editor.org/info/rfc7252>>.

Acknowledgements

Matthias Kovatsch forced the author to make up his mind about this. Ari Keranen forced him to write it up, then, and created a convincing use case of Content-Format-Strings. John Mattsson alerted us to a mistake. Alexey Melnikov suggested to revive this draft after a year of dormancy.

Author's Address

Carsten Bormann
Universität Bremen TZI
Postfach 330440
D-28359 Bremen
Germany

Phone: [+49-421-218-63921](tel:+49-421-218-63921)

Email: cabo@tzi.org