On Media-Types, Content-Types, and related terminology
draft-bormann-core-media-content-type-format-01

Abstract

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

This memo is an attempt at clearing it up.

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 January 9, 2020.

Copyright Notice

Copyright (c) 2019 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.
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:

\begin{verbatim}
type-name = reg-name
subtype-name = reg-name
\end{verbatim}

\begin{verbatim}
reg-name = 1*127reg-name-chars
reg-name-chars = ALPHA / DIGIT / "!" / "\#" / "$" / 
"\&" / "." / 
"\" / 
"\^" / 
"_"
\end{verbatim}

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:

\begin{verbatim}
Media-Type-Name = type-name "/" subtype-name
\end{verbatim}

For the purposes of this memo, we define:

\begin{verbatim}
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.
\end{verbatim}
(This leaves the term "Media Type" for the actual specification that is registered under the Media-Type-Name.)

3. Content-Type

Media types 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 1: 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 conversion, as well as some other cobwebs, we define the conventional textual representation of a Content-Type as:
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:

\[
\text{Content-Format} = 1^*\text{DIGIT} \\
\text{Content-Format-String} = \text{Content-Type} ["@" \text{content-coding}]
\]

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 sometime abbreviated as "mt", and Content-Types as "ct". We do not propose 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, while the normative ABNF of this memo is left-aligned.)

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",

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

10.2. Informative References


[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,

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.

Author’s Address

Carsten Bormann
Universitaet Bremen T2I
Postfach 330440
Bremen D-28359
Germany

Phone: +49-421-218-63921
Email: cabo@tzi.org