HTTP Working Group
Internet-Draft
Obsoletes: 5987 (if approved)
Intended status: Standards Track
Expires: September 5, 2016

J. Reschke
greenbytes

March 4, 2016

Indicating Character Encoding and Language for HTTP Header Field Parameters
draft-ietf-httpbis-rfc5987bis-01

Abstract

By default, message header field parameters in Hypertext Transfer Protocol (HTTP) messages cannot carry characters outside the ISO-8859-1 character set. RFC 2231 defines an encoding mechanism for use in Multipurpose Internet Mail Extensions (MIME) headers. This document specifies an encoding suitable for use in HTTP header fields that is compatible with a profile of the encoding defined in RFC 2231.

Editorial Note (To be removed by RFC Editor before publication)

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

Working Group information can be found at <http://httpwg.github.io/>; source code and issues list for this draft can be found at <https://github.com/httpwg/http-extensions>.

The changes in this draft are summarized in Appendix C.

Status of This Memo

This Internet-Draft will expire on September 5, 2016.

Copyright Notice

Table of Contents

1. Introduction

By default, message header field parameters in HTTP ([RFC2616]) messages cannot carry characters outside the ISO-8859-1 coded character set ([ISO-8859-1]). RFC 2231 ([RFC2231]) defines an encoding mechanism for use in MIME headers. This document specifies an encoding suitable for use in HTTP header fields that is compatible with a profile of the encoding defined in RFC 2231.

This document obsoletes [RFC5987] and moves it to "historic" status; the changes are summarized in Appendix A.

> Note: in the remainder of this document, RFC 2231 is only referenced for the purpose of explaining the choice of features that were adopted; they are therefore purely informative.
>
> Note: this encoding does not apply to message payloads transmitted over HTTP, such as when using the media type "multipart/form-data" ([RFC2388]).

2. Notational Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

This specification uses the ABNF (Augmented Backus-Naur Form) notation defined in [RFC5234]. The following core rules are included by reference, as defined in [RFC5234], Appendix B.1: ALPHA (letters), DIGIT (decimal 0-9), HEXDIG (hexadecimal 0-9/A-F/a-f), and LWSP (linear whitespace).

This specification uses terminology defined in [RFC6365], namely: "character encoding scheme" (below abbreviated to "character encoding"), "charset" and "coded character set".

Note that this differs from RFC 2231, which uses the term "character set" for "character encoding scheme".

3. Comparison to RFC 2231 and Definition of the Encoding

RFC 2231 defines several extensions to MIME. The sections below discuss if and how they apply to HTTP header fields.

In short:

- o Parameter Continuations aren't needed (Section 3.1),

- Character Encoding and Language Information are useful, therefore a simple subset is specified (Section 3.2), and

- Language Specifications in Encoded Words aren't needed (Section 3.3).

3.1. Parameter Continuations

Section 3 of [RFC2231] defines a mechanism that deals with the length limitations that apply to MIME headers. These limitations do not apply to HTTP ([RFC7231], Appendix A.6).

Thus, parameter continuations are not part of the encoding defined by this specification.

3.2. Parameter Value Character Encoding and Language Information

Section 4 of [RFC2231] specifies how to embed language information into parameter values, and also how to encode non-ASCII characters, dealing with restrictions both in MIME and HTTP header field parameters.

However, RFC 2231 does not specify a mandatory-to-implement character encoding, making it hard for senders to decide which encoding to use. Thus, recipients implementing this specification MUST support the "UTF-8" character encoding [RFC3629].

Furthermore, RFC 2231 allows the character encoding information to be left out. The encoding defined by this specification does not allow that.

3.2.1. Definition

The syntax for parameters is defined in Section 3.6 of [RFC2616] (with RFC 2616 implied LWS translated to RFC 5234 LWSP):

```
  parameter     = attribute LWSP "=" LWSP value

  attribute     = token
  value         = token / quoted-string

  quoted-string = <quoted-string, see [RFC7230], Section 3.2.6>
  token         = <token, see [RFC7230], Section 3.2.6>
```

In order to include character encoding and language information, this specification modifies the RFC 2616 grammar to be:

```
     parameter     = reg-parameter / ext-parameter

     reg-parameter = parmname LWSP "=" LWSP value

     ext-parameter = parmname "*" LWSP "=" LWSP ext-value

     parmname      = 1*attr-char

     ext-value     = charset  "'" [ language ] "'" value-chars
                   ; like RFC 2231's <extended-initial-value>
                   ; (see [RFC2231], Section 7)

     charset       = "UTF-8" / mime-charset

     mime-charset  = 1*mime-charsetc
     mime-charsetc = ALPHA / DIGIT
                   / "!" / "#" / "$" / "%" / "&"
                   / "+" / "-" / "^" / "_" / "`"
                   / "{" / "}" / "~"
                   ; as <mime-charset> in Section 2.3 of [RFC2978]
                   ; except that the single quote is not included
                   ; SHOULD be registered in the IANA charset registry

     language      = <Language-Tag, see [RFC5646], Section 2.1>

     value-chars   = *( pct-encoded / attr-char )

     pct-encoded   = "%" HEXDIG HEXDIG
                   ; see [RFC3986], Section 2.1

     attr-char     = ALPHA / DIGIT
                   / "!" / "#" / "$" / "&" / "+" / "-" / "."
                   / "^" / "_" / "`" / "|" / "~"
                   ; token except ( "*" / "'" / "%" )
```

Thus, a parameter is either a regular parameter (reg-parameter), as previously defined in Section 3.6 of [RFC2616], or an extended parameter (ext-parameter).

Extended parameters are those where the left-hand side of the assignment ends with an asterisk character.

The value part of an extended parameter (ext-value) is a token that consists of three parts: the REQUIRED character encoding name (charset), the OPTIONAL language information (language), and a character sequence representing the actual value (value-chars), separated by single quote characters. Note that both character encoding names and language tags are restricted to the US-ASCII coded

character set, and are matched case-insensitively (see [RFC2978], Section 2.3 and [RFC5646], Section 2.1.1).

Inside the value part, characters not contained in attr-char are encoded into an octet sequence using the specified character encoding. That octet sequence is then percent-encoded as specified in Section 2.1 of [RFC3986].

Producers MUST use the "UTF-8" ([RFC3629]) character encoding. Extension character encodings (mime-charset) are reserved for future use.

> Note: recipients should be prepared to handle encoding errors, such as malformed or incomplete percent escape sequences, or non-decodable octet sequences, in a robust manner. This specification does not mandate any specific behavior, for instance, the following strategies are all acceptable:
>
> * ignoring the parameter,
>
> * stripping a non-decodable octet sequence,
>
> * substituting a non-decodable octet sequence by a replacement character, such as the Unicode character U+FFFD (Replacement Character).

3.2.2. Historical Notes

The RFC 7230 token production ([RFC7230], Section 3.2.6) differs from the production used in RFC 2231 (imported from Section 5.1 of [RFC2045]) in that curly braces ("{" and "}") are excluded. Thus, these two characters are excluded from the attr-char production as well.

The <mime-charset> ABNF defined here differs from the one in Section 2.3 of [RFC2978] in that it does not allow the single quote character (see also RFC Errata ID 1912 [Err1912]). In practice, no character encoding names using that character have been registered at the time of this writing.

For backwards compatibility with RFC 2231, the encoding defined by this specification deviates from common parameter syntax in that the quoted-string notation is not allowed. Implementations using generic parser components might not be able to detect the use of quoted-string notation and thus might accept that format, although invalid, as well.

[RFC5987] did require support for ISO-8859-1, too; for compatibility

with legacy code, recipients are encouraged to support this encoding as well.

3.2.3. Examples

Non-extended notation, using "token":

```
foo: bar; title=Economy
```

Non-extended notation, using "quoted-string":

```
foo: bar; title="US-$ rates"
```

Extended notation, using the Unicode character U+00A3 ("£", POUND SIGN):

```
foo: bar; title*=utf-8'en'%C2%A3%20rates
```

Note: the Unicode pound sign character U+00A3 was encoded into the octet sequence C2 A3 using the UTF-8 character encoding, then percent-encoded. Also, note that the space character was encoded as %20, as it is not contained in attr-char.

Extended notation, using the Unicode characters U+00A3 ("£", POUND SIGN) and U+20AC ("€", EURO SIGN):

```
foo: bar; title*=UTF-8''%c2%a3%20and%20%e2%82%ac%20rates
```

Note: the Unicode pound sign character U+00A3 was encoded into the octet sequence C2 A3 using the UTF-8 character encoding, then percent-encoded. Likewise, the Unicode euro sign character U+20AC was encoded into the octet sequence E2 82 AC, then percent-encoded. Also note that HEXDIG allows both lowercase and uppercase characters, so recipients must understand both, and that the language information is optional, while the character encoding is not.

3.3. Language Specification in Encoded Words

Section 5 of [RFC2231] extends the encoding defined in [RFC2047] to also support language specification in encoded words. RFC 2616, the now-obsolete HTTP/1.1 specification, did refer to RFC 2047 ([RFC2616], Section 2.2). However, it wasn't clear to which header field it applied. Consequently, the current revision of the HTTP/1.1 specification has deprecated use of the encoding forms defined in RFC 2047 (see Section 3.2.4 of [RFC7230]).

Thus, this specification does not include this feature.

4. Guidelines for Usage in HTTP Header Field Definitions

Specifications of HTTP header fields that use the extensions defined in Section 3.2 ought to clearly state that. A simple way to achieve this is to normatively reference this specification, and to include the ext-value production into the ABNF for that header field.

For instance:

```
  foo-header  = "foo" LWSP ":" LWSP token ";" LWSP title-param
  title-param = "title" LWSP "=" LWSP value
              / "title*" LWSP "=" LWSP ext-value
  ext-value   = <see RFC 5987, Section 3.2>
```

> Note: The Parameter Value Continuation feature defined in Section 3 of [RFC2231] makes it impossible to have multiple instances of extended parameters with identical parmname components, as the processing of continuations would become ambiguous. Thus, specifications using this extension are advised to disallow this case for compatibility with RFC 2231.

> Note: This specification does not automatically assign a new interpretation to parameter names ending in an asterisk. As pointed out above, it's up to the specification for the non-extended parameter to "opt in" to the syntax defined here. That being said, some existing implementations are known to automatically switch to the use of this notation when a parameter name ends with an asterisk, thus using parameter names ending in an asterisk for something else is likely to cause interoperability problems.

4.1. When to Use the Extension

Section 4.2 of [RFC2277] requires that protocol elements containing human-readable text are able to carry language information. Thus, the ext-value production ought to be always used when the parameter value is of textual nature and its language is known.

Furthermore, the extension ought to also be used whenever the parameter value needs to carry characters not present in the US-ASCII ([USASCII]) coded character set (note that it would be unacceptable to define a new parameter that would be restricted to a subset of the Unicode character set).

4.2. Error Handling

Header field specifications need to define whether multiple instances of parameters with identical parmname components are allowed, and how

they should be processed. This specification suggests that a parameter using the extended syntax takes precedence. This would allow producers to use both formats without breaking recipients that do not understand the extended syntax yet.

Example:

```
  foo: bar; title="EURO exchange rates";
            title*=utf-8''%e2%82%ac%20exchange%20rates
```

In this case, the sender provides an ASCII version of the title for legacy recipients, but also includes an internationalized version for recipients understanding this specification -- the latter obviously ought to prefer the new syntax over the old one.

> Note: at the time of this writing, many implementations failed to ignore the form they do not understand, or prioritize the ASCII form although the extended syntax was present.

5. Security Considerations

The format described in this document makes it possible to transport non-ASCII characters, and thus enables character "spoofing" scenarios, in which a displayed value appears to be something other than it is.

Furthermore, there are known attack scenarios relating to decoding UTF-8.

See Section 10 of [RFC3629] for more information on both topics.

In addition, the extension specified in this document makes it possible to transport multiple language variants for a single parameter, and such use might allow spoofing attacks, where different language versions of the same parameter are not equivalent. Whether this attack is useful as an attack depends on the parameter specified.

6. IANA Considerations

There are no IANA Considerations related to this specification.

7. References

7.1. Normative References

[RFC2119]   Bradner, S., "Key words for use in RFCs to Indicate
              Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/

RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[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, <http://www.rfc-editor.org/info/rfc2616>.

[RFC2978] Freed, N. and J. Postel, "IANA Charset Registration Procedures", BCP 19, RFC 2978, DOI 10.17487/RFC2978, October 2000, <http://www.rfc-editor.org/info/rfc2978>.

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, DOI 10.17487/RFC3629, November 2003, <http://www.rfc-editor.org/info/rfc3629>.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <http://www.rfc-editor.org/info/rfc3986>.

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

[RFC5646] Phillips, A., Ed. and M. Davis, Ed., "Tags for Identifying Languages", BCP 47, RFC 5646, DOI 10.17487/RFC5646, September 2009, <http://www.rfc-editor.org/info/rfc5646>.

[RFC7230] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, DOI 10.17487/RFC7230, June 2014, <http://www.rfc-editor.org/info/rfc7230>.

[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, <http://www.rfc-editor.org/info/rfc7231>.

[USASCII] American National Standards Institute, "Coded Character Set -- 7-bit American Standard Code for Information Interchange", ANSI X3.4, 1986.

7.2. Informative References

[Err1912] RFC Errata, "Errata ID 1912, RFC 2978", <http://www.rfc-editor.org>.

[ISO-8859-1] International Organization for Standardization, "Information technology -- 8-bit single-byte coded graphic character sets -- Part 1: Latin alphabet No. 1", ISO/IEC 8859-1:1998, 1998.

[RFC2045] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, DOI 10.17487/RFC2045, November 1996, <http://www.rfc-editor.org/info/rfc2045>.

[RFC2047] Moore, K., "MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text", RFC 2047, DOI 10.17487/RFC2047, November 1996, <http://www.rfc-editor.org/info/rfc2047>.

[RFC2231] Freed, N. and K. Moore, "MIME Parameter Value and Encoded Word Extensions: Character Sets, Languages, and Continuations", RFC 2231, DOI 10.17487/RFC2231, November 1997, <http://www.rfc-editor.org/info/rfc2231>.

[RFC2277] Alvestrand, H., "IETF Policy on Character Sets and Languages", BCP 18, RFC 2277, DOI 10.17487/RFC2277, January 1998, <http://www.rfc-editor.org/info/rfc2277>.

[RFC2388] Masinter, L., "Returning Values from Forms: multipart/form-data", RFC 2388, DOI 10.17487/RFC2388, August 1998, <http://www.rfc-editor.org/info/rfc2388>.

[RFC5987] Reschke, J., "Character Set and Language Encoding for Hypertext Transfer Protocol (HTTP) Header Field Parameters", RFC 5987, DOI 10.17487/RFC5987, August 2010, <http://www.rfc-editor.org/info/rfc5987>.

[RFC5988] Nottingham, M., "Web Linking", RFC 5988, DOI 10.17487/RFC5988, October 2010, <http://www.rfc-editor.org/info/rfc5988>.

[RFC6266] Reschke, J., "Use of the Content-Disposition Header Field in the Hypertext Transfer Protocol (HTTP)", RFC 6266, DOI 10.17487/RFC6266, June 2011,

<http://www.rfc-editor.org/info/rfc6266>.

[RFC6365] Hoffman, P. and J. Klensin, "Terminology Used in Internationalization in the IETF", BCP 166, RFC 6365, DOI 10.17487/RFC6365, September 2011, <http://www.rfc-editor.org/info/rfc6365>.

Appendix A. Changes from RFC 5987

This section summarizes the changes compared to [RFC5987]:

- The document title was changed to "Indicating Character Encoding and Language for HTTP Header Field Parameters".

- The requirement to support the "ISO-8859-1" encoding was removed.

Appendix B. Implementation Report

The encoding defined in this document currently is used for two different HTTP header fields:

- "Content-Disposition", defined in [RFC6266], and

- "Link", defined in [RFC5988].

As the encoding is a profile/clarification of the one defined in [RFC2231] in 1997, many user agents already supported it for use in "Content-Disposition" when [RFC5987] got published.

Since the publication of [RFC5987], three more popular desktop user agents have added support for this encoding; see <http://purl.org/NET/http/content-disposition-tests#encoding-2231-char> for details. At this time, the current versions of all major desktop user agents support it.

Note that the implementation in Internet Explorer 9 does not support the ISO-8859-1 character encoding; this document revision acknowledges that UTF-8 is sufficient for expressing all code points, and removes the requirement to support ISO-8859-1.

The "Link" header field, on the other hand, was only recently specified in [RFC5988]. At the time of this writing, no shipping User Agent except Firefox supported the "title*" parameter (starting with release 15).

Appendix C. Change Log (to be removed by RFC Editor before publication)

C.1. Since RFC5987

Only editorial changes for the purpose of starting the revision process (obs5987).

C.2. Since draft-reschke-rfc5987bis-00

Resolved issues "iso-8859-1" and "title" (title simplified). Added and resolved issue "historic5987".

C.3. Since draft-reschke-rfc5987bis-01

Added issues "httpbis", "parmsyntax", "terminology" and "valuesyntax". Closed issue "impls".

C.4. Since draft-reschke-rfc5987bis-02

Resolved issue "terminology".

C.5. Since draft-reschke-rfc5987bis-03

In Section 3.2, pull historical notes into a separate subsection. Resolved issues "valuesyntax" and "parmsyntax".

C.6. Since draft-reschke-rfc5987bis-04

Update status of Firefox support in HTTP Link Header field.

C.7. Since draft-reschke-rfc5987bis-05

Update status of Firefox support in HTTP Link Header field.

C.8. Since draft-reschke-rfc5987bis-06

Update status with respect to Safari 6.

Started work on update with respect to RFC 723x.

C.9. Since draft-ietf-httpbis-rfc5987bis-00

Editorial changes; introducing non-ASCII characters into author's address, acknowledgements, and examples.

Appendix D. Acknowledgements

Thanks to Martin Dürst and Frank Ellermann for help figuring out ABNF details, to Graham Klyne and Alexey Melnikov for general review, to Chris Newman for pointing out an RFC 2231 incompatibility, and to Benjamin Carlyle, Roar Lauritzsen, Eric Lawrence, and James Manger for implementer's feedback.

Author's Address

Julian F. Reschke
greenbytes GmbH
Hafenweg 16
Münster, NW 48155
Germany

EMail: julian.reschke@greenbytes.de
URI: http://greenbytes.de/tech/webdav/