An Encoding Parameter for HTTP Basic Authentication
draft-reschke-basicauth-enc-05

Abstract

The "Basic" authentication scheme defined in RFC 2617 does not properly define how to treat non-ASCII characters. This has lead to a situation where user agent implementations disagree, and servers make different assumptions based on the locales they are running in. There is little interoperability for the non-ASCII characters in the ISO-8859-1 character set, and even less interoperability for any characters beyond that.

This document defines a backwards-compatible extension to "Basic", specifying the server’s character encoding expectation, using a new authentication scheme parameter.

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

Distribution of this document is unlimited. Although this is not a work item of the HTTPbis Working Group, comments should be sent to the Hypertext Transfer Protocol (HTTP) mailing list at ietf-http-wg@w3.org [1], which may be joined by sending a message with subject "subscribe" to ietf-http-wg-request@w3.org [2].

Discussions of the HTTPbis Working Group are archived at <http://lists.w3.org/Archives/Public/ietf-http-wg/>.

XML versions, latest edits and the issues list for this document are available from <http://greenbytes.de/tech/webdav/#draft-reschke-basicauth-enc>.

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 http://datatracker.ietf.org/drafts/current/.
Table of Contents

1. Introduction ................................................. 4
2. Notational Conventions .................................... 4
3. The ‘accept-charset’ auth-param ............................ 4
4. Examples ..................................................... 5
5. Security Considerations .................................... 5
6. IANA Considerations ....................................... 5
7. Acknowledgements .......................................... 5
8. References .................................................. 6
   8.1. Normative References .................................. 6
   8.2. Informative References ................................. 7
Appendix A. Deployment Considerations ......................... 7
   A.1. User Agents ............................................. 7
       A.1.1. Alternative approach ............................... 7
   A.2. Origin Servers .......................................... 8
Appendix B. FAQ (to be removed by RFC Editor before publication) ................................. 8
   B.1. Why not simply switch the default encoding to UTF-8? .... 8
   B.2. What about Digest? ...................................... 8
   B.3. Will existing UAs ignore the parameter? ............... 8
Appendix C. Change Log (to be removed by RFC Editor before publication) ......................... 8
   C.1. Since draft-reschke-basicauth-enc-00 ................... 9
   C.2. Since draft-reschke-basicauth-enc-01 ................... 9
   C.3. Since draft-reschke-basicauth-enc-02 ................... 9
   C.4. Since draft-reschke-basicauth-enc-03 ................... 9
   C.5. Since draft-reschke-basicauth-enc-04 ................... 9
Appendix D. Resolved issues (to be removed by RFC Editor before publication) .................. 9
   D.1. sentparam ................................................. 9
   D.2. paramname ............................................... 10
Appendix E. Open issues (to be removed by RFC Editor prior to publication) .................... 10
   E.1. edit ...................................................... 10
   E.2. unorm ................................................... 10
   E.3. terminology .............................................. 10
1. Introduction

The "Basic" authentication scheme defined in Section 2 of [RFC2617] does not properly define how to treat non-ASCII characters ([USASCII]): it uses the Base64 ([RFC4648], Section 4) encoding of the concatenation of username, separator character, and password without stating which character encoding to use.

This has lead to a situation where user agent implementations disagree, and servers make different assumptions based on the locales they are running in. There is little interoperability for the non-ASCII characters in the ISO-8859-1 character set ([USASCII], [ISO-8859-1]), and even less interoperability for any characters beyond that.

This document defines a backwards-compatible extension to "Basic", specifying the server’s character encoding expectation, using a new auth-param for use in the Proxy-Authenticate and WWW-Authenticate header fields, as defined in [draft-ietf-httpbis-p7-auth].

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].

3. The ‘accept-charset’ auth-param

In challenges, servers MAY use the "accept-charset" authentication parameter (case-insensitive) to express the character encoding they expect the user agent to use.

The only allowed value is "UTF-8", to be matched case-insensitively (see [RFC2978], Section 2.3), indicating that the server expects the UTF-8 character encoding to be used ([RFC3629]).

Other values are reserved for future use.

Note: The ‘accept-charset’ parameter cannot be included when sending credentials (e.g. in the Authorization or Proxy-Authorization header fields), as the "Basic" scheme uses a single base64 token for credentials ('b64token' syntax), not a parameter list ('#auth-param' syntax); see Section 2.1 of [draft-ietf-httpbis-p7-auth].
4. Examples

In the example below, the server prompts for authentication in the "foo" realm, using Basic authentication, with a preference for the UTF-8 character encoding:

WWW-Authenticate: Basic realm="foo", accept-charset="UTF-8"

Note that the parameter value can be either a token or a quoted string; in this case the server chose to use the quoted-string notation.

The user’s name is "test", and his password is the string "123" followed by the Unicode character U+00A3 (POUND SIGN). Following Section 1.2 of [RFC2617], but using the character encoding UTF-8, the user-pass, converted to a sequence of octets, is:

't' 'e' 's' 't' ':' '1' '2' '3' pound
74 65 73 74 3A 31 32 33 C2 A3

Encoding this octet sequence in Base64 ([RFC4648], Section 4) yields:

dGVzdDoxMjPCow==

Thus the Authorization header field would be:

Authorization: Basic dGVzdDoxMjPCow==

Or, for proxy authentication:

Proxy-Authorization: Basic dGVzdDoxMjPCow==

5. Security Considerations

This document does not introduce any new security considerations beyond those defined for the "Basic" authentication scheme ([RFC2617], Section 4), and those applicable to the handling of UTF-8 ([RFC3629], Section 10).

6. IANA Considerations

There are no IANA Considerations related to this specification.

7. Acknowledgements

The internationalisation problem has been reported as a Mozilla bug back in the year 2000 (see <https://bugzilla.mozilla.org/show_bug.cgi?id=41489> and also the
more recent <https://bugzilla.mozilla.org/show_bug.cgi?id=656213>).
It was Andrew Clover’s idea to address it using a new auth-param.

Thanks to Bjoern Hoehrmann, Amos Jeffries, James Manger, and Martin Thomson for providing feedback on this document.

8. References

8.1. Normative References


8.2. Informative References


URIs

[1] <mailto:ietf-http-wg@w3.org>

[2] <mailto:ietf-http-wg-request@w3.org?subject=subscribe>

Appendix A. Deployment Considerations

A.1. User Agents

User agents not implementing this specification should continue to work as before, ignoring the new parameter.

User agents which already default to the UTF-8 encoding implement this specification by definition. Note that some user agents also have different defaults depending on whether the request originates from page navigation as opposed to a script-driven request using XMLHttpRequest [XHR].

Other user agents can keep their default behavior, and switch to UTF-8 when seeing the new parameter.

A.1.1. Alternative approach

On the other hand, the strategy below may already improve the user-visible behavior today:

- In the first authentication request, choose the character encoding based on the user’s credentials: if they do not need any characters outside the ISO-8859-1 character set, default to ISO-8859-1, otherwise use UTF-8.
If the first attempt failed and the encoding used was ISO-8859-1, retry once with UTF-8 encoding instead.

Note that there’s a risk if the site blocks an account after multiple login failures (for instance, when it doesn’t reset the counter after a successful login).

A.2. Origin Servers

Origin servers that do not support non-ASCII characters in credentials do not require any changes.

Origin servers that need to support non-ASCII characters, but can’t use the UTF-8 encoding will not be affected; they will continue to function as well as before.

Finally, origin servers that need to support non-ASCII characters and can use the UTF-8 encoding can opt in as described above. In the worst case, they’ll continue to see either broken credentials or no credentials at all (depending on how legacy clients handle characters they can not encode).

Appendix B. FAQ (to be removed by RFC Editor before publication)

B.1. Why not simply switch the default encoding to UTF-8?

There are sites in use today that default to a locale encoding, such as ISO-8859-1, and expect user agents to use that encoding. These sites will break if the user agent uses a different encoding, such as UTF-8.

B.2. What about Digest?

Although the solution proposed in this document may be applicable to "Digest" as well, any attempt to update this scheme may be an uphill battle hard to win.

B.3. Will existing UAs ignore the parameter?

It appears they will. See <http://greenbytes.de/tech/tc/httpauth/#simplebasicnewparam1> and <http://greenbytes.de/tech/tc/httpauth/#simplebasicnewparam2>.

Appendix C. Change Log (to be removed by RFC Editor before publication)
C.1. Since draft-reschke-basicauth-enc-00

Add and close issues "credparam" and "paramcase". Rewrite the deployment considerations.

C.2. Since draft-reschke-basicauth-enc-01

Note more recent Mozilla bugzilla entry; add behavior of existing UAs to FAQ (with pointer to test cases).

C.3. Since draft-reschke-basicauth-enc-02

Add and resolve issue "xhrutf8".

C.4. Since draft-reschke-basicauth-enc-03

Add and resolve issue "proxy".

C.5. Since draft-reschke-basicauth-enc-04

Add and resolve issues "paramname" and "sentparam". Add issues "terminology" and "unorm". Update HTTPbis reference.

Appendix D. Resolved issues (to be removed by RFC Editor before publication)

Issues that were either rejected or resolved in this version of this document.

D.1. sentparam

Type: change

<http://lists.w3.org/Archives/Public/ietf-http-wg/2012JanMar/0299.html>

James.H.Manger@team.telstra.com (2011-01-30): The text about not including the 'encoding' parameter when sending the password is a bit confusing [section 3]. (...) My guess is that the spec intended to say that including the encoding information *would* be useful, but it cannot be added easily. This is a good illustration of the 3rd dot point from "2.3.1 Considerations for new Authentication Schemes" [draft-ietf-httpbis-p7-auth-18#section-2.3.1]: "b64token ... can only be used once ... future extensions will be impossible".

D.2. paramname

Type: change

<http://lists.w3.org/Archives/Public/ietf-http-wg/2012JanMar/0302.html>

derhoermi@gmx.net (2012-01-30): ... (in part due to the name, 'useUTF8' or 'use-utf-8="yes" or some such would have been clearer)

Resolution (2012-03-11): Switch to "accept-charset", so this is similar to the HTML form attribute.

Appendix E. Open issues (to be removed by RFC Editor prior to publication)

E.1. edit

Type: edit

julian.reschke@greenbytes.de (2010-08-11): Umbrella issue for editorial fixes/enhancements.

E.2. unorm

Type: edit

julian.reschke@greenbytes.de (2012-02-02): We need a statement about unicode normalization forms.

E.3. terminology

Type: edit

julian.reschke@greenbytes.de (2012-02-02): Try to be consistent with the terminology defined in RFC 6365.

Author’s Address

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

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