

HTTPAuth Working Group  
Internet-Draft  
Obsoletes: [2617](#) (if approved)  
Intended status: Standards Track  
Expires: June 22, 2015

J. Reschke  
greenbytes  
December 19, 2014

**The 'Basic' HTTP Authentication Scheme  
draft-ietf-httpauth-basicauth-update-04**

Abstract

This document defines the "Basic" Hypertext Transfer Protocol (HTTP) Authentication Scheme, which transmits credentials as userid/password pairs, obfuscated by the use of Base64 encoding.

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

Discussion of this draft takes place on the HTTPAuth working group mailing list ([http-auth@ietf.org](mailto:http-auth@ietf.org)), which is archived at [<http://www.ietf.org/mail-archive/web/http-auth/current/maillist.html>](http://www.ietf.org/mail-archive/web/http-auth/current/maillist.html).

XML versions, latest edits and the issues list for this document are available from [<http://greenbytes.de/tech/webdav/#draft-ietf-httpauth-basicauth-update>](http://greenbytes.de/tech/webdav/#draft-ietf-httpauth-basicauth-update).

The changes in this draft are summarized in [Appendix C.5](#).

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/>.

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 June 22, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the

document authors. All rights reserved.

This document is subject to [BCP 78](http://trustee.ietf.org/license-info) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://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.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.



## Table of Contents

<a href="#">1.</a>	<a href="#">Introduction</a>	<a href="#">4</a>
<a href="#">1.1.</a>	<a href="#">Notational Conventions</a>	<a href="#">4</a>
<a href="#">1.1.1.</a>	<a href="#">Syntax Notation</a>	<a href="#">4</a>
<a href="#">2.</a>	<a href="#">The 'Basic' Authentication Scheme</a>	<a href="#">4</a>
<a href="#">2.1.</a>	<a href="#">The 'charset' auth-param</a>	<a href="#">6</a>
<a href="#">2.2.</a>	<a href="#">Re-using Credentials</a>	<a href="#">8</a>
<a href="#">3.</a>	<a href="#">Internationalization Considerations</a>	<a href="#">8</a>
<a href="#">4.</a>	<a href="#">Security Considerations</a>	<a href="#">9</a>
<a href="#">5.</a>	<a href="#">IANA Considerations</a>	<a href="#">10</a>
<a href="#">6.</a>	<a href="#">Acknowledgements</a>	<a href="#">10</a>
<a href="#">7.</a>	<a href="#">References</a>	<a href="#">11</a>
<a href="#">7.1.</a>	<a href="#">Normative References</a>	<a href="#">11</a>
<a href="#">7.2.</a>	<a href="#">Informative References</a>	<a href="#">11</a>
<a href="#">Appendix A.</a>	<a href="#">Changes from <a href="#">RFC 2617</a></a>	<a href="#">12</a>
<a href="#">Appendix B.</a>	<a href="#">Deployment Considerations for the 'charset' Parameter</a>	<a href="#">12</a>
<a href="#">B.1.</a>	<a href="#">User Agents</a>	<a href="#">12</a>
<a href="#">B.2.</a>	<a href="#">Origin Servers</a>	<a href="#">13</a>
<a href="#">B.3.</a>	<a href="#">Why not simply switch the default encoding to UTF-8?</a>	<a href="#">13</a>
<a href="#">Appendix C.</a>	<a href="#">Change Log (to be removed by RFC Editor before publication)</a>	<a href="#">13</a>
<a href="#">C.1.</a>	<a href="#">Since <a href="#">RFC 2617</a></a>	<a href="#">13</a>
<a href="#">C.2.</a>	<a href="#">Since <a href="#">draft-ietf-httpauth-basicauth-update-00</a></a>	<a href="#">13</a>
<a href="#">C.3.</a>	<a href="#">Since <a href="#">draft-ietf-httpauth-basicauth-update-01</a></a>	<a href="#">14</a>
<a href="#">C.4.</a>	<a href="#">Since <a href="#">draft-ietf-httpauth-basicauth-update-02</a></a>	<a href="#">14</a>
<a href="#">C.5.</a>	<a href="#">Since <a href="#">draft-ietf-httpauth-basicauth-update-03</a></a>	<a href="#">14</a>



## 1. Introduction

This document defines the "Basic" Hypertext Transfer Protocol (HTTP) Authentication Scheme, which transmits credentials as userid/password pairs, obfuscated by the use of Base64 encoding (HTTP authentication schemes are defined in [\[RFC7235\]](#)).

This scheme is not considered to be a secure method of user authentication unless used in conjunction with some external secure system such as TLS (Transport Layer Security, [\[RFC5246\]](#)), as the user name and password are passed over the network as cleartext.

The "Basic" scheme previously was defined in [Section 2 of \[RFC2617\]](#). This document updates the definition, and also addresses internationalization issues by introducing the "charset" authentication parameter ([Section 2.1](#)).

Other documents updating [RFC 2617](#) are "Hypertext Transfer Protocol (HTTP/1.1): Authentication" ([\[RFC7235\]](#), defining the authentication framework) and "HTTP Digest Access Authentication" ([\[DIGEST\]](#), updating the definition of the "Digest" authentication scheme). Taken together, these three documents obsolete [RFC 2617](#).

### 1.1. 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 [\[RFC2119\]](#).

#### 1.1.1. Syntax Notation

This specification uses the Augmented Backus-Naur Form (ABNF) notation of [\[RFC5234\]](#).

The terms protection space and realm are defined in [Section 2.2 of \[RFC7235\]](#).

The terms (character) repertoire and character encoding scheme are defined in [Section 2 of \[RFC6365\]](#).

## 2. The 'Basic' Authentication Scheme

The "Basic" authentication scheme is based on the model that the client needs to authenticate itself with a user-ID and a password for each protection space ("realm"). The realm value is an opaque string which can only be compared for equality with other realms on that server. The server will service the request only if it can validate the user-ID and password for the protection space applying to the



requested resource.

The "Basic" authentication scheme utilizes the Authentication Framework as follows:

In challenges:

- o the scheme name is "Basic"
- o the authentication parameter "realm" is REQUIRED ([\[RFC7235\], Section 2.2](#))
- o the authentication parameter "charset" is OPTIONAL (see [Section 2.1](#))
- o no other authentication parameters are defined -- unknown parameters MUST be ignored by recipients, and new parameters can only be defined by revising this specification

Note that both scheme and parameter names are matched case-insensitively.

For credentials, the "token68" syntax defined in [Section 2.1 of \[RFC7235\]](#) is used. The value is computed based on user-id and password as defined below.

Upon receipt of a request for a URI within the protection space that lacks credentials, the server can reply with a challenge using the 401 (Unauthorized) status code ([\[RFC7235\], Section 3.1](#)) and the WWW-Authenticate header field ([\[RFC7235\], Section 4.1](#)).

For instance:

```
HTTP/1.1 401 Unauthorized
Date: Mon, 04 Feb 2014 16:50:53 GMT
WWW-Authenticate: Basic realm="WallyWorld"
```

...where "WallyWorld" is the string assigned by the server to identify the protection space.

A proxy can respond with a similar challenge using the 407 (Proxy Authentication Required) status code ([\[RFC7235\], Section 3.2](#)) and the Proxy-Authenticate header field ([\[RFC7235\], Section 4.3](#)).

To receive authorization, the client





1. obtains the userid and password from the user,
2. constructs the user-pass by concatenating userid, a single colon (":") character, and the password,
3. encodes the user-pass into an octet sequence (see below for a discussion of character encoding schemes),
4. and obtains the basic-credentials by encoding this octet sequence using base64 ([\[RFC4648\]](#), [Section 4](#)) into a sequence of US-ASCII characters ([\[RFC0020\]](#)).

The original definition of this authentication scheme failed to specify the character encoding scheme used to convert the user-pass into an octet sequence. In practice, most implementations chose either a local-specific encoding such as ISO-8859-1 ([\[ISO-8859-1\]](#)), or UTF-8 ([\[RFC3629\]](#)). For backwards compatibility reasons, this specification continues to leave the default encoding undefined, as long as it is compatible with US-ASCII (mapping any US-ASCII character to a single octet matching the US-ASCII character code).

The userid and password MUST NOT contain any control characters (see "CTL" in [Appendix B.1 of \[RFC5234\]](#)).

Furthermore, a userid containing a colon character is invalid, as recipients will split the user-pass at the first occurrence of a colon character. Note that many user agents however will accept a colon in userid, thereby producing a user-pass string that recipients will likely treat in a way not intended by the user.

If the user agent wishes to send the userid "Aladdin" and password "open sesame", it would use the following header field:

```
Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==
```

### **[2.1](#). The 'charset' auth-param**

In challenges, servers can use the "charset" authentication parameter to indicate the character encoding scheme they expect the user agent to use when generating "user-pass" (a sequence of octets). This information is purely advisory.

The only allowed value is "UTF-8", to be matched case-insensitively (see [\[RFC2978\]](#), [Section 2.3](#)). It indicates that the server expects character data to be converted to Unicode Normalization Form C ("NFC", see [Section 3 of \[RFC5198\]](#)) and to be encoded into octets using the UTF-8 character encoding scheme ([\[RFC3629\]](#)).



For the userid, recipients MUST support all characters defined in the "UsernameCasePreserved" profile defined in in Section 3.3 of [\[PRECIS\]](#), with the exception of the colon (":") character.

For the password, recipients MUST support all characters defined in the "OpaqueString" profile defined in in Section 4.2 of [\[PRECIS\]](#).

Other values are reserved for future use.

Note: The 'charset' is only defined on challenges, as "Basic" uses a single token for credentials ('token68' syntax), thus the credentials syntax isn't extensible.

Note: The name 'charset' has been chosen for consistency with [Section 2.1.1 of \[RFC2831\]](#). A better name would have been 'accept-charset', as it is not about the message it appears in, but the server's expectation.

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 scheme:

```
WWW-Authenticate: Basic realm="foo", 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 the password is the string "123" followed by the Unicode character U+00A3 (POUND SIGN). Using the character encoding scheme UTF-8, the user-pass becomes:

```
'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==
```



## **2.2. Re-using Credentials**

Given the absolute URI ([\[RFC3986\]](#), [Section 4.3](#)) of an authenticated request, the authentication scope of that request is obtained by removing all characters after the last slash ("/") character. A client SHOULD assume that resources identified by URIs with a prefix-match of the authentication scope are also within the protection space specified by the realm value of the that authenticated request.

A client MAY preemptively send the corresponding Authorization header field with requests for resources in that space without receipt of another challenge from the server. Similarly, when a client sends a request to a proxy, it may reuse a userid and password in the Proxy-Authorization header field without receiving another challenge from the proxy server.

For example, given an authenticated request to:

```
http://example.com/docs/index.html
```

...requests to the URIs below could use the known credentials:

```
http://example.com/docs/  
http://example.com/docs/test.doc  
http://example.com/docs/?page=1
```

...while the URIs

```
http://example.com/other/  
https://example.com/docs/
```

would be considered to be outside the authentication scope.

Note that a URI can be part of multiple authentication scopes (such as "http://example.com/" and "http://example.com/docs/"). This specification does not define which of these should be treated with higher priority.

## **3. Internationalization Considerations**

User names or passwords containing characters outside the US-ASCII character repertoire will cause interoperability issues, unless both communication partners agree on what character encoding scheme is to be used. Servers can use the new 'charset' parameter ([Section 2.1](#)) to indicate a preference of "UTF-8", increasing the probability that clients will switch to that encoding.

The "realm" parameter carries data that can be considered textual,



however [[RFC7235](#)] does not define a way to reliably transport non-US-ASCII characters. This is a known issue that would need to be addressed in that specification.

#### 4. Security Considerations

The Basic authentication scheme is not a secure method of user authentication, nor does it in any way protect the entity, which is transmitted in cleartext across the physical network used as the carrier. HTTP does not prevent the addition of enhancements (such as schemes to use one-time passwords) to Basic authentication.

The most serious flaw in Basic authentication is that it results in the cleartext transmission of the user's password over the physical network. Many other authentication schemes address this problem.

Because Basic authentication involves the cleartext transmission of passwords it SHOULD NOT be used (without enhancements such as HTTPS [[RFC2818](#)]) to protect sensitive or valuable information.

A common use of Basic authentication is for identification purposes -- requiring the user to provide a user name and password as a means of identification, for example, for purposes of gathering accurate usage statistics on a server. When used in this way it is tempting to think that there is no danger in its use if illicit access to the protected documents is not a major concern. This is only correct if the server issues both user name and password to the users and in particular does not allow the user to choose his or her own password. The danger arises because naive users frequently reuse a single password to avoid the task of maintaining multiple passwords.

If a server permits users to select their own passwords, then the threat is not only unauthorized access to documents on the server but also unauthorized access to any other resources on other systems that the user protects with the same password. Furthermore, in the server's password database, many of the passwords may also be users' passwords for other sites. The owner or administrator of such a system could therefore expose all users of the system to the risk of unauthorized access to all those sites if this information is not maintained in a secure fashion. This raises both security and privacy concerns ([[RFC6973](#)]). If the same username and password combination is in use to access other accounts, such as an email or health portal account, personal information could be exposed.

Basic Authentication is also vulnerable to spoofing by counterfeit servers. If a user can be led to believe that he is connecting to a host containing information protected by Basic authentication when, in fact, he is connecting to a hostile server or gateway, then the





attacker can request a password, store it for later use, and feign an error. This type of attack is not possible with Digest Authentication. Server implementers SHOULD guard against the possibility of this sort of counterfeiting by gateways or CGI scripts. In particular it is very dangerous for a server to simply turn over a connection to a gateway. That gateway can then use the persistent connection mechanism to engage in multiple transactions with the client while impersonating the original server in a way that is not detectable by the client.

The use of the UTF-8 character encoding scheme introduces additional security considerations; see [Section 10 of \[RFC3629\]](#) for more information.

## 5. IANA Considerations

IANA maintains the registry of HTTP Authentication Schemes ([RFC7235]) at <http://www.iana.org/assignments/http-authschemes>.

The entry for the "Basic" Authentication Scheme shall be updated with a pointer to this specification.

## 6. Acknowledgements

This specification takes over the definition of the "Basic" HTTP Authentication Scheme, previously defined in [RFC 2617](#). We thank John Franks, Phillip M. Hallam-Baker, Jeffery L. Hostetler, Scott D. Lawrence, Paul J. Leach, Ari Luotonen, and Lawrence C. Stewart for their work on that specification, from which significant amounts of text were borrowed. See [Section 6 of \[RFC2617\]](#) for further acknowledgements.

The internationalization problem with respect to the character encoding scheme used for user-pass has been reported as a Mozilla bug back in the year 2000 (see [https://bugzilla.mozilla.org/show\\_bug.cgi?id=41489](https://bugzilla.mozilla.org/show_bug.cgi?id=41489) and also the more recent [https://bugzilla.mozilla.org/show\\_bug.cgi?id=656213](https://bugzilla.mozilla.org/show_bug.cgi?id=656213)). It was Andrew Clover's idea to address it using a new auth-param.

We also thank the members of the HTTPAuth Working Group and other reviewers, namely Stephen Farrell, Bjoern Hoehrmann, Kari Hurtta, Amos Jeffries, Benjamin Kaduk, James Manger, Kathleen Moriarty, Yaron Sheffer, Michael Sweet, and Martin Thomson for feedback on this revision.

## 7. References



### **7.1. Normative References**

- [PRECIS] Saint-Andre, P. and A. Melnikov, "Preparation, Enforcement, and Comparison of Internationalized Strings Representing Usernames and Passwords", [draft-ietf-precis-saslprepbis-12](#) (work in progress), December 2014.
- [RFC0020] Cerf, V., "ASCII format for network interchange", [RFC 20](#), October 1969.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC2978] Freed, N. and J. Postel, "IANA Charset Registration Procedures", [BCP 19](#), [RFC 2978](#), October 2000.
- [RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, [RFC 3629](#), November 2003.
- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, [RFC 3986](#), January 2005.
- [RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", [RFC 4648](#), October 2006.
- [RFC5198] Klensin, J. and M. Padlipsky, "Unicode Format for Network Interchange", [RFC 5198](#), March 2008.
- [RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, [RFC 5234](#), January 2008.
- [RFC6365] Hoffman, P. and J. Klensin, "Terminology Used in Internationalization in the IETF", [BCP 166](#), [RFC 6365](#), September 2011.
- [RFC7235] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Authentication", [RFC 7235](#), June 2014.

### **7.2. Informative References**

- [DIGEST] Shekh-Yusef, R., Ed., Ahrens, D., and S. Bremer, "HTTP Digest Access Authentication", [draft-ietf-httpauth-digest-09](#) (work in progress), December 2014.



- [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.
- [RFC2617] Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A., and L. Stewart, "HTTP Authentication: Basic and Digest Access Authentication", [RFC 2617](#), June 1999.
- [RFC2818] Rescorla, E., "HTTP Over TLS", [RFC 2818](#), May 2000.
- [RFC2831] Leach, P. and C. Newman, "Using Digest Authentication as a SASL Mechanism", [RFC 2831](#), May 2000.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", [RFC 5246](#), August 2008.
- [RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", [RFC 6973](#), July 2013.
- [RFC7231] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", [RFC 7231](#), June 2014.

## [Appendix A.](#) Changes from [RFC 2617](#)

The scheme definition has been rewritten to be consistent with newer specifications such as [\[RFC7235\]](#).

The new authentication parameter "charset" has been added. It is purely advisory, so existing implementations do not need to change, unless they want to take advantage of the additional information which previously wasn't available.

## [Appendix B.](#) Deployment Considerations for the 'charset' Parameter

### [B.1.](#) User Agents

User agents not implementing 'charset' will continue to work as before, ignoring the new parameter.

User agents which already default to the UTF-8 encoding implement 'charset' by definition.



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

## **B.2. Origin Servers**

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

Origin servers that need to support non-US-ASCII characters, but cannot use the UTF-8 character encoding scheme will not be affected; they will continue to function as well or as badly as before.

Finally, origin servers that need to support non-US-ASCII characters and can use the UTF-8 character encoding scheme 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).

## **B.3. Why not simply switch the default encoding to UTF-8?**

There are sites in use today that default to a local character encoding scheme, such as ISO-8859-1 ([\[ISO-8859-1\]](#)), and expect user agents to use that encoding. Authentication on these sites will stop to work if the user agent switches to a different encoding, such as UTF-8.

Note that sites might even inspect the User-Agent header field ([\[RFC7231\]](#), [Section 5.5.3](#)) to decide what character encoding scheme to expect from the client. Therefore they might support UTF-8 for some user agents, but default to something else for others. User agents in the latter group will have to continue to do what they do today until the majority of these servers have been upgraded to always use UTF-8.

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

### **C.1. Since [RFC 2617](#)**

This draft acts as a baseline for tracking subsequent changes to the specification. As such, it extracts the definition of "Basic", plus the related Security Considerations, and also adds the IANA registration of the scheme. Changes to the actual definition will be made in subsequent drafts.

### **C.2. Since [draft-ietf-httpauth-basicauth-update-00](#)**

Fixed Base64 reference to point to an actual definition of Base64.





Update HTTPbis and Digest references.

Note that this spec, together with HTTPbis P7 and the Digest update, obsoletes [RFC 2617](#).

Rewrote text about authentication parameters and their extensibility.

Pulled in the definition of the "charset" parameter.

Removed a misleading statement about userids potentially being case-sensitive, as the same is true for passwords.

Added TODOs with respect to path matching, and colons in userids.

### **C.3. Since [draft-ietf-httpauth-basicauth-update-01](#)**

Minor improvements on Security Considerations.

Update Digest reference.

Rewrite scheme definition as algorithm rather than pseudo-ABNF.

Add a note about colons in userid.

Attempt to explain authentication scopes.

### **C.4. Since [draft-ietf-httpauth-basicauth-update-02](#)**

Reference [draft-ietf-precis-saslprepbis](#) for the set of characters that need to be supported in userids and passwords.

### **C.5. Since [draft-ietf-httpauth-basicauth-update-03](#)**

Update reference for [draft-ietf-precis-saslprepbis](#) (which renames "Password" to "OpaqueString").

Mention HTTPS as enhancement for securing the transmission of credentials.

Update DIGEST reference and change it to informative.

Use [RFC 20](#) as reference for ASCII.



## Author's Address

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

EMail: [julian.reschke@greenbytes.de](mailto:julian.reschke@greenbytes.de)

URI: <http://greenbytes.de/tech/webdav/>