

How many HTTP verbs are there?



I count 9 [HTTP request methods](#):

6

GET
HEAD
POST
PUT
DELETE
CONNECT
OPTIONS
TRACE
PATCH



1

The above from: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods>

Is that it? will this ever change?

http

methods

request

protocols

edited Jan 19 at 19:14



[Basil Bourque](#)

125k 34 423 592

asked Dec 31 '16 at 19:27



[Purplejacket](#)

634 12 25

Note: I know that formally these are "HTTP request methods" but I also see them referred to as "verbs", as the link above describes. — [Purplejacket](#)
Jan 16 '17 at 19:28

2 Answers



Registry

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).



+50

- ACL
- BASELINE-CONTROL
- BIND
- CHECKIN
- CHECKOUT
- CONNECT
- COPY
- DELETE
- GET
- HEAD
- LABEL
- LINK
- LOCK
- MERGE
- MKACTIVITY
- MKCALENDAR
- MKCOL
- MKREDIRECTREF
- MKWORKSPACE
- MOVE
- OPTIONS
- ORDERPATCH
- PATCH
- POST
- PRI
- PROPFIND

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

- REBIND
- REPORT
- SEARCH
- TRACE
- UNBIND
- UNCHECKOUT
- UNLINK
- UNLOCK
- UPDATE
- UPDATEREDIRECTREF
- VERSION-CONTROL

HTTP 1.0

[HTTP](#) 1.0 defined three methods (“verbs”):

- **GET**
... retrieve whatever information ... is identified by the Request-URI...
- **POST**
... to request that the destination server accept the entity enclosed in the request as a new subordinate of the resource identified by the Request-URI in the Request-Line... Posting a message to a bulletin board, newsgroup, mailing list ... Providing a block of data ... Extending a database through an append operation ...
- **HEAD**
... identical to GET except that the server MUST NOT return a message-body in the response ... for obtaining metainformation about the entity implied by the request without transferring the entity-body itself...

HTTP 1.1

HTTP 1.1 is officially defined in [RFC 2068](#). This spec added five more methods.

- **OPTIONS**

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

- **PUT**
...requests that the enclosed entity be stored under the supplied Request-URI. If ... already existing resource, the enclosed entity SHOULD be considered as a modified version of the one residing on the origin server...
- **DELETE**
...delete the resource identified by the Request-URI...
- **TRACE**
...loop- back of the request message...
- **CONNECT**
...for use with a proxy that can dynamically switch to being a tunnel (e.g. SSL tunneling...

HTTP Extensions

Other protocols extend HTTP to define additional methods/verbs.

- [PATCH](#)
 - Applies partial modifications to a resource
 - Defined by [RFC 5789](#)
- [WebDAV](#) specifies seven more methods:
 - PROPFIND
 - PROPPATCH
 - MKCOL
 - COPY
 - MOVE
 - LOCK
 - UNLOCK

HTTP/2

[HTTP/2](#) is defined in [RFC 7540](#). [Section 3.5](#) defines a `PRI` method.

- **PRI**

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

Prognostication

will this ever change?

Not likely.

Given the wide use of Web [RPC](#) and [SOAP](#), and now the rising popularity of [RESTful](#) services bringing new life to the existing basic verbs, there is little need to devise new verbs at the HTTP level. Where people need their own domain-specific meaningful verbs, they can embed within the message being delivered via HTTP.

I expect we'll not see more HTTP methods become popular any time soon.

edited Jan 19 at 19:21

answered Dec 31 '16 at 20:30



[Basil Bourque](#)

125k 34 423 592

Regarding HTTP/2: <http2.github.io> says "HTTP methods, status codes and semantics are the same", so it seems HTTP/2 doesn't introduce any new request methods ("verbs"), though in the spec I do see mention of a PRI method: "This method is never used by an actual client. This method will appear to be used when an HTTP/1.1 server or intermediary attempts to parse an HTTP/2 connection preface." – [Purplejacket](#) Jan 1 '17 at 2:17

Is WebDAV actually a thing? I don't see it in practice. Admittedly my background is with webapps and REST APIs. – [Purplejacket](#) Jan 1 '17 at 2:39

1 WebDAV is used quite a lot in content management, also (through CalDAV) in calendaring. – [Julian Reschke](#) Jan 2 '17 at 8:30

Will there be many more HTTP request methods in the future, or is this process winding down, so that what we have now is basically it? – [Purplejacket](#) Jan 16 '17 at 22:30

@Purplejacket By design, anyone can devise any HTTP methods they want at any time. Any small scale usage by mutual agreement between a small number of parties is inconsequential. Whether something as significant and wide-spread like WebDAV will come along is difficult to predict. The Registry was last updated 2015-05-19. With [REST](#) technology having given new life to the existing methods, I expect we'll not see more methods become popular any time soon. – [Basil Bourque](#) Jan 16 '17 at 23:43

See the spec:

2 "Additional methods, outside the scope of this specification, have been standardized for use in HTTP. All such methods ought to be

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

And the [IANA registry](#) contains many more.

edited Jan 2 '17 at 8:34



[Basil Bourque](#)

125k 34 423 592

answered Jan 1 '17 at 8:26



[Julian Reschke](#)

29k 5 64 71

-
- 1 I find the IANA registry at iana.org/assignments/http-methods/http-methods.xhtml – this mentions 39 methods as of January 2017. Quite a few of these are from WebDAV or extensions to WebDAV. I do notice LINK and UNLINK which are related to HTTP 1.1, PRI related to HTTP/2. – [Purplejacket](#) Jan 1 '17 at 22:30
-

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).