wget http://mirrors.oit.uci.edu/centos/7.9.2009/isos/x86\_64/CentOS-7-x86\_64-DVD-2009.iso

(base) mbpro2:httpd-2.4.46 moyer$ ./configure --help

`configure' configures this package to adapt to many kinds of systems.

Usage: ./configure [OPTION]... [VAR=VALUE]...

To assign environment variables (e.g., CC, CFLAGS...), specify them as

VAR=VALUE. See below for descriptions of some of the useful variables.

Defaults for the options are specified in brackets.

Configuration:

-h, --help display this help and exit

--help=short display options specific to this package

--help=recursive display the short help of all the included packages

-V, --version display version information and exit

-q, --quiet, --silent do not print `checking ...' messages

--cache-file=FILE cache test results in FILE [disabled]

-C, --config-cache alias for `--cache-file=config.cache'

-n, --no-create do not create output files

--srcdir=DIR find the sources in DIR [configure dir or `..']

Installation directories:

--prefix=PREFIX install architecture-independent files in PREFIX

[/usr/local/apache2]

--exec-prefix=EPREFIX install architecture-dependent files in EPREFIX

[PREFIX]

By default, `make install' will install all the files in

`/usr/local/apache2/bin', `/usr/local/apache2/lib' etc. You can specify

an installation prefix other than `/usr/local/apache2' using `--prefix',

for instance `--prefix=$HOME'.

For better control, use the options below.

Fine tuning of the installation directories:

--bindir=DIR user executables [EPREFIX/bin]

--sbindir=DIR system admin executables [EPREFIX/sbin]

--libexecdir=DIR program executables [EPREFIX/libexec]

--sysconfdir=DIR read-only single-machine data [PREFIX/etc]

--sharedstatedir=DIR modifiable architecture-independent data [PREFIX/com]

--localstatedir=DIR modifiable single-machine data [PREFIX/var]

--runstatedir=DIR modifiable per-process data [LOCALSTATEDIR/run]

--libdir=DIR object code libraries [EPREFIX/lib]

--includedir=DIR C header files [PREFIX/include]

--oldincludedir=DIR C header files for non-gcc [/usr/include]

--datarootdir=DIR read-only arch.-independent data root [PREFIX/share]

--datadir=DIR read-only architecture-independent data [DATAROOTDIR]

--infodir=DIR info documentation [DATAROOTDIR/info]

--localedir=DIR locale-dependent data [DATAROOTDIR/locale]

--mandir=DIR man documentation [DATAROOTDIR/man]

--docdir=DIR documentation root [DATAROOTDIR/doc/PACKAGE]

--htmldir=DIR html documentation [DOCDIR]

--dvidir=DIR dvi documentation [DOCDIR]

--pdfdir=DIR pdf documentation [DOCDIR]

--psdir=DIR ps documentation [DOCDIR]

System types:

--build=BUILD configure for building on BUILD [guessed]

--host=HOST cross-compile to build programs to run on HOST [BUILD]

--target=TARGET configure for building compilers for TARGET [HOST]

Optional Features:

--disable-option-checking ignore unrecognized --enable/--with options

--disable-FEATURE do not include FEATURE (same as --enable-FEATURE=no)

--enable-FEATURE[=ARG] include FEATURE [ARG=yes]

--enable-layout=LAYOUT

--enable-dtrace Enable DTrace probes

--enable-hook-probes Enable APR hook probes

--enable-exception-hook Enable fatal exception hook

--enable-load-all-modules

Load all modules

--enable-maintainer-mode

Turn on debugging and compile time warnings and load

all compiled modules

--enable-debugger-mode Turn on debugging and compile time warnings and turn

off optimization

--enable-pie Build httpd as a Position Independent Executable

--enable-modules=MODULE-LIST

Space-separated list of modules to enable | "all" |

"most" | "few" | "none" | "reallyall"

--enable-mods-shared=MODULE-LIST

Space-separated list of shared modules to enable |

"all" | "most" | "few" | "reallyall"

--enable-mods-static=MODULE-LIST

Space-separated list of static modules to enable |

"all" | "most" | "few" | "reallyall"

--disable-authn-file file-based authentication control

--enable-authn-dbm DBM-based authentication control

--enable-authn-anon anonymous user authentication control

--enable-authn-dbd SQL-based authentication control

--enable-authn-socache Cached authentication control

--disable-authn-core core authentication module

--disable-authz-host host-based authorization control

--disable-authz-groupfile

'require group' authorization control

--disable-authz-user 'require user' authorization control

--enable-authz-dbm DBM-based authorization control

--enable-authz-owner 'require file-owner' authorization control

--enable-authz-dbd SQL based authorization and Login/Session support

--disable-authz-core core authorization provider vector module

--enable-authnz-ldap LDAP based authentication

--enable-authnz-fcgi FastCGI authorizer-based authentication and

authorization

--disable-access-compat mod\_access compatibility

--disable-auth-basic basic authentication

--enable-auth-form form authentication

--enable-auth-digest RFC2617 Digest authentication

--enable-allowmethods restrict allowed HTTP methods

--enable-isapi isapi extension support

--enable-file-cache File cache

--enable-cache dynamic file caching. At least one storage

management module (e.g. mod\_cache\_disk) is also

necessary.

--enable-cache-disk disk caching module

--enable-cache-socache shared object caching module

--enable-socache-shmcb shmcb small object cache provider

--enable-socache-dbm dbm small object cache provider

--enable-socache-memcache

memcache small object cache provider

--enable-socache-redis redis small object cache provider

--enable-socache-dc distcache small object cache provider

--enable-so DSO capability. This module will be automatically

enabled unless you build all modules statically.

--enable-watchdog Watchdog module

--enable-macro Define and use macros in configuration files

--enable-dbd Apache DBD Framework

--enable-bucketeer buckets manipulation filter. Useful only for

developers and testing purposes.

--enable-dumpio I/O dump filter

--enable-echo ECHO server

--enable-example-hooks Example hook callback handler module

--enable-case-filter Example uppercase conversion filter

--enable-case-filter-in Example uppercase conversion input filter

--enable-example-ipc Example of shared memory and mutex usage

--enable-buffer Filter Buffering

--enable-data RFC2397 data encoder

--enable-ratelimit Output Bandwidth Limiting

--disable-reqtimeout Limit time waiting for request from client

--enable-ext-filter external filter module

--enable-request Request Body Filtering

--enable-include Server Side Includes

--disable-filter Smart Filtering

--enable-reflector Reflect request through the output filter stack

--enable-substitute response content rewrite-like filtering

--enable-sed filter request and/or response bodies through sed

--disable-charset-lite character set translation. Enabled by default only

on EBCDIC systems.

--enable-charset-lite character set translation. Enabled by default only

on EBCDIC systems.

--enable-deflate Deflate transfer encoding support

--enable-xml2enc i18n support for markup filters

--enable-proxy-html Fix HTML Links in a Reverse Proxy

--enable-brotli Brotli compression support

--enable-http HTTP protocol handling. The http module is a basic

one that enables the server to function as an HTTP

server. It is only useful to disable it if you want

to use another protocol module instead. Don't

disable this module unless you are really sure what

you are doing. Note: This module will always be

linked statically.

--disable-mime mapping of file-extension to MIME. Disabling this

module is normally not recommended.

--enable-ldap LDAP caching and connection pooling services

--disable-log-config logging configuration. You won't be able to log

requests to the server without this module.

--enable-log-debug configurable debug logging

--enable-log-forensic forensic logging

--enable-logio input and output logging

--enable-lua Apache Lua Framework

--enable-luajit Enable LuaJit Support

--disable-env clearing/setting of ENV vars

--enable-mime-magic automagically determining MIME type

--enable-cern-meta CERN-type meta files

--enable-expires Expires header control

--disable-headers HTTP header control

--enable-ident RFC 1413 identity check

--enable-usertrack user-session tracking

--enable-unique-id per-request unique ids

--disable-setenvif basing ENV vars on headers

--disable-version determining httpd version in config files

--enable-remoteip translate header contents to an apparent client

remote\_ip

--enable-proxy Apache proxy module

--enable-proxy-connect Apache proxy CONNECT module. Requires

--enable-proxy.

--enable-proxy-ftp Apache proxy FTP module. Requires --enable-proxy.

--enable-proxy-http Apache proxy HTTP module. Requires --enable-proxy.

--enable-proxy-fcgi Apache proxy FastCGI module. Requires

--enable-proxy.

--enable-proxy-scgi Apache proxy SCGI module. Requires --enable-proxy.

--enable-proxy-uwsgi Apache proxy UWSGI module. Requires --enable-proxy.

--enable-proxy-fdpass Apache proxy to Unix Daemon Socket module. Requires

--enable-proxy.

--enable-proxy-wstunnel Apache proxy Websocket Tunnel module. Requires

--enable-proxy.

--enable-proxy-ajp Apache proxy AJP module. Requires --enable-proxy.

--enable-proxy-balancer Apache proxy BALANCER module. Requires

--enable-proxy.

--enable-proxy-express mass reverse-proxy module. Requires --enable-proxy.

--enable-proxy-hcheck reverse-proxy health-check module. Requires

--enable-proxy and --enable-watchdog.

--enable-session session module

--enable-session-cookie session cookie module

--enable-session-crypto session crypto module

--enable-session-dbd session dbd module

--enable-slotmem-shm slotmem provider that uses shared memory

--enable-slotmem-plain slotmem provider that uses plain memory

--enable-ssl SSL/TLS support (mod\_ssl)

--enable-ssl-staticlib-deps

link mod\_ssl with dependencies of OpenSSL's static

libraries (as indicated by "pkg-config --static").

Must be specified in addition to --enable-ssl.

--enable-optional-hook-export

example optional hook exporter

--enable-optional-hook-import

example optional hook importer

--enable-optional-fn-import

example optional function importer

--enable-optional-fn-export

example optional function exporter

--enable-dialup rate limits static files to dialup modem speeds

--enable-static-support Build a statically linked version of the support

binaries

--enable-static-htpasswd

Build a statically linked version of htpasswd

--enable-static-htdigest

Build a statically linked version of htdigest

--enable-static-rotatelogs

Build a statically linked version of rotatelogs

--enable-static-logresolve

Build a statically linked version of logresolve

--enable-static-htdbm Build a statically linked version of htdbm

--enable-static-ab Build a statically linked version of ab

--enable-static-checkgid

Build a statically linked version of checkgid

--enable-static-htcacheclean

Build a statically linked version of htcacheclean

--enable-static-httxt2dbm

Build a statically linked version of httxt2dbm

--enable-static-fcgistarter

Build a statically linked version of fcgistarter

--enable-http2 HTTP/2 protocol handling in addition to HTTP

protocol handling. Implemented by mod\_http2. This

module requires a libnghttp2 installation. See

--with-nghttp2 on how to manage non-standard

locations. This module is usually linked shared and

requires loading.

--enable-nghttp2-staticlib-deps

link mod\_http2 with dependencies of libnghttp2's

static libraries (as indicated by "pkg-config

--static"). Must be specified in addition to

--enable-http2.

--enable-proxy-http2 HTTP/2 proxy module. This module requires a

libnghttp2 installation. See --with-nghttp2 on how

to manage non-standard locations. Also requires

--enable-proxy.

--enable-md Managed Domain handling

--enable-jansson-staticlib-deps

link mod\_md with dependencies of libjansson's static

libraries (as indicated by "pkg-config --static").

Must be specified in addition to --enable-md.

--enable-curl-staticlib-deps

link mod\_md with dependencies of libcurl's static

libraries (as indicated by "pkg-config --static").

Must be specified in addition to --enable-md.

--enable-lbmethod-byrequests

Apache proxy Load balancing by request counting

--enable-lbmethod-bytraffic

Apache proxy Load balancing by traffic counting

--enable-lbmethod-bybusyness

Apache proxy Load balancing by busyness

--enable-lbmethod-heartbeat

Apache proxy Load balancing from Heartbeats

--enable-mpms-shared=MPM-LIST

Space-separated list of MPM modules to enable for

dynamic loading. MPM-LIST=list | "all"

--enable-unixd unix specific support

--enable-privileges Per-virtualhost Unix UserIDs and enhanced security

for Solaris

--enable-systemd Systemd support

--enable-heartbeat Generates Heartbeats

--enable-heartmonitor Collects Heartbeats

--enable-dav WebDAV protocol handling. --enable-dav also enables

mod\_dav\_fs

--disable-status process/thread monitoring

--disable-autoindex directory listing

--enable-asis as-is filetypes

--enable-info server information

--enable-suexec set uid and gid for spawned processes

--enable-cgid CGI scripts. Enabled by default with threaded MPMs

--enable-cgi CGI scripts. Enabled by default with non-threaded

MPMs

--enable-dav-fs DAV provider for the filesystem. --enable-dav also

enables mod\_dav\_fs.

--enable-dav-lock DAV provider for generic locking

--enable-vhost-alias mass virtual hosting module

--enable-negotiation content negotiation

--disable-dir directory request handling

--enable-imagemap server-side imagemaps

--enable-actions Action triggering on requests

--enable-speling correct common URL misspellings

--enable-userdir mapping of requests to user-specific directories

--disable-alias mapping of requests to different filesystem parts

--enable-rewrite rule based URL manipulation

--enable-suexec-capabilities

Use Linux capability bits not setuid root suexec

--enable-v4-mapped Allow IPv6 sockets to handle IPv4 connections

Optional Packages:

--with-PACKAGE[=ARG] use PACKAGE [ARG=yes]

--without-PACKAGE do not use PACKAGE (same as --with-PACKAGE=no)

--with-included-apr Use bundled copies of APR/APR-Util

--with-apr=PATH prefix for installed APR or the full path to

apr-config

--with-apr-util=PATH prefix for installed APU or the full path to

apu-config

--with-pcre=PATH Use external PCRE library

--with-port=PORT Port on which to listen (default is 80)

--with-sslport=SSLPORT Port on which to securelisten (default is 443)

--with-distcache=PATH Distcache installation directory

--with-z=PATH use a specific zlib library

--with-libxml2=PATH location for libxml2

--with-brotli=PATH Brotli installation directory

--with-lua=PATH Path to the Lua 5.3/5.2/5.1 prefix

--with-ssl=PATH OpenSSL installation directory

--with-nghttp2=PATH nghttp2 installation directory

--with-jansson=PATH jansson installation directory

--with-curl=PATH curl installation directory

--with-mpm=MPM Choose the process model for Apache to use by

default. MPM={event|worker|prefork|winnt} This will

be statically linked as the only available MPM

unless --enable-mpms-shared is also specified.

--with-module=module-type:module-file

Enable module-file in the modules/<module-type>

directory.

--with-program-name alternate executable name

--with-suexec-bin Path to suexec binary

--with-suexec-caller User allowed to call SuExec

--with-suexec-userdir User subdirectory

--with-suexec-docroot SuExec root directory

--with-suexec-uidmin Minimal allowed UID

--with-suexec-gidmin Minimal allowed GID

--with-suexec-logfile Set the logfile

--with-suexec-syslog Use syslog for suexec logging

--with-suexec-safepath Set the safepath

--with-suexec-umask umask for suexec'd process

Some influential environment variables:

CC C compiler command

CFLAGS C compiler flags

LDFLAGS linker flags, e.g. -L<lib dir> if you have libraries in a

nonstandard directory <lib dir>

LIBS libraries to pass to the linker, e.g. -l<library>

CPPFLAGS (Objective) C/C++ preprocessor flags, e.g. -I<include dir> if

you have headers in a nonstandard directory <include dir>

CPP C preprocessor

Use these variables to override the choices made by `configure' or to help

it to find libraries and programs with nonstandard names/locations.

Report bugs to the package provider.

(base) mbpro2:httpd-2.4.46 moyer$

10 Status Code Definitions

Each Status-Code is described below, including a description of which method(s) it can follow and any metainformation required in the response.

10.1 Informational 1xx

This class of status code indicates a provisional response, consisting only of the Status-Line and optional headers, and is terminated by an empty line. There are no required headers for this class of status code. Since HTTP/1.0 did not define any 1xx status codes, servers MUST NOT send a 1xx response to an HTTP/1.0 client except under experimental conditions.

A client MUST be prepared to accept one or more 1xx status responses prior to a regular response, even if the client does not expect a 100 (Continue) status message. Unexpected 1xx status responses MAY be ignored by a user agent.

Proxies MUST forward 1xx responses, unless the connection between the proxy and its client has been closed, or unless the proxy itself requested the generation of the 1xx response. (For example, if a

proxy adds a "Expect: 100-continue" field when it forwards a request, then it need not forward the corresponding 100 (Continue) response(s).)

10.1.1 100 Continue

The client SHOULD continue with its request. This interim response is used to inform the client that the initial part of the request has been received and has not yet been rejected by the server. The client SHOULD continue by sending the remainder of the request or, if the request has already been completed, ignore this response. The server MUST send a final response after the request has been completed. See section [8.2.3](https://www.w3.org/Protocols/rfc2616/rfc2616-sec8.html#sec8.2.3) for detailed discussion of the use and handling of this status code.

10.1.2 101 Switching Protocols

The server understands and is willing to comply with the client's request, via the Upgrade message header field (section 14.42), for a change in the application protocol being used on this connection. The server will switch protocols to those defined by the response's Upgrade header field immediately after the empty line which terminates the 101 response.

The protocol SHOULD be switched only when it is advantageous to do so. For example, switching to a newer version of HTTP is advantageous over older versions, and switching to a real-time, synchronous protocol might be advantageous when delivering resources that use such features.

10.2 Successful 2xx

This class of status code indicates that the client's request was successfully received, understood, and accepted.

10.2.1 200 OK

The request has succeeded. The information returned with the response is dependent on the method used in the request, for example:

GET an entity corresponding to the requested resource is sent in the response;

HEAD the entity-header fields corresponding to the requested resource are sent in the response without any message-body;

POST an entity describing or containing the result of the action;

TRACE an entity containing the request message as received by the end server.

10.2.2 201 Created

The request has been fulfilled and resulted in a new resource being created. The newly created resource can be referenced by the URI(s) returned in the entity of the response, with the most specific URI for the resource given by a Location header field. The response SHOULD include an entity containing a list of resource characteristics and location(s) from which the user or user agent can choose the one most appropriate. The entity format is specified by the media type given in the Content-Type header field. The origin server MUST create the resource before returning the 201 status code. If the action cannot be carried out immediately, the server SHOULD respond with 202 (Accepted) response instead.

A 201 response MAY contain an ETag response header field indicating the current value of the entity tag for the requested variant just created, see section [14.19](https://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.19).

10.2.3 202 Accepted

The request has been accepted for processing, but the processing has not been completed. The request might or might not eventually be acted upon, as it might be disallowed when processing actually takes place. There is no facility for re-sending a status code from an asynchronous operation such as this.

The 202 response is intentionally non-committal. Its purpose is to allow a server to accept a request for some other process (perhaps a batch-oriented process that is only run once per day) without requiring that the user agent's connection to the server persist until the process is completed. The entity returned with this response SHOULD include an indication of the request's current status and either a pointer to a status monitor or some estimate of when the user can expect the request to be fulfilled.

10.2.4 203 Non-Authoritative Information

The returned metainformation in the entity-header is not the definitive set as available from the origin server, but is gathered from a local or a third-party copy. The set presented MAY be a subset or superset of the original version. For example, including local annotation information about the resource might result in a superset of the metainformation known by the origin server. Use of this response code is not required and is only appropriate when the response would otherwise be 200 (OK).

10.2.5 204 No Content

The server has fulfilled the request but does not need to return an entity-body, and might want to return updated metainformation. The response MAY include new or updated metainformation in the form of entity-headers, which if present SHOULD be associated with the requested variant.

If the client is a user agent, it SHOULD NOT change its document view from that which caused the request to be sent. This response is primarily intended to allow input for actions to take place without causing a change to the user agent's active document view, although any new or updated metainformation SHOULD be applied to the document currently in the user agent's active view.

The 204 response MUST NOT include a message-body, and thus is always terminated by the first empty line after the header fields.

10.2.6 205 Reset Content

The server has fulfilled the request and the user agent SHOULD reset the document view which caused the request to be sent. This response is primarily intended to allow input for actions to take place via user input, followed by a clearing of the form in which the input is given so that the user can easily initiate another input action. The response MUST NOT include an entity.

10.2.7 206 Partial Content

The server has fulfilled the partial GET request for the resource. The request MUST have included a Range header field (section 14.35) indicating the desired range, and MAY have included an If-Range header field (section [14.27](https://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.27)) to make the request conditional.

The response MUST include the following header fields:

- Either a Content-Range header field (section 14.16) indicating

the range included with this response, or a multipart/byteranges

Content-Type including Content-Range fields for each part. If a

Content-Length header field is present in the response, its

value MUST match the actual number of OCTETs transmitted in the

message-body.

- Date

- ETag and/or Content-Location, if the header would have been sent

in a 200 response to the same request

- Expires, Cache-Control, and/or Vary, if the field-value might

differ from that sent in any previous response for the same

variant

If the 206 response is the result of an If-Range request that used a strong cache validator (see section 13.3.3), the response SHOULD NOT include other entity-headers. If the response is the result of an If-Range request that used a weak validator, the response MUST NOT include other entity-headers; this prevents inconsistencies between cached entity-bodies and updated headers. Otherwise, the response MUST include all of the entity-headers that would have been returned with a 200 (OK) response to the same request.

A cache MUST NOT combine a 206 response with other previously cached content if the ETag or Last-Modified headers do not match exactly, see [13.5.4](https://www.w3.org/Protocols/rfc2616/rfc2616-sec13.html#sec13.5.4).

A cache that does not support the Range and Content-Range headers MUST NOT cache 206 (Partial) responses.

10.3 Redirection 3xx

This class of status code indicates that further action needs to be taken by the user agent in order to fulfill the request. The action required MAY be carried out by the user agent without interaction with the user if and only if the method used in the second request is GET or HEAD. A client SHOULD detect infinite redirection loops, since such loops generate network traffic for each redirection.

Note: previous versions of this specification recommended a

maximum of five redirections. Content developers should be aware

that there might be clients that implement such a fixed

limitation.

10.3.1 300 Multiple Choices

The requested resource corresponds to any one of a set of representations, each with its own specific location, and agent- driven negotiation information (section 12) is being provided so that the user (or user agent) can select a preferred representation and redirect its request to that location.

Unless it was a HEAD request, the response SHOULD include an entity containing a list of resource characteristics and location(s) from which the user or user agent can choose the one most appropriate. The entity format is specified by the media type given in the Content- Type header field. Depending upon the format and the capabilities of

the user agent, selection of the most appropriate choice MAY be performed automatically. However, this specification does not define any standard for such automatic selection.

If the server has a preferred choice of representation, it SHOULD include the specific URI for that representation in the Location field; user agents MAY use the Location field value for automatic redirection. This response is cacheable unless indicated otherwise.

10.3.2 301 Moved Permanently

The requested resource has been assigned a new permanent URI and any future references to this resource SHOULD use one of the returned URIs. Clients with link editing capabilities ought to automatically re-link references to the Request-URI to one or more of the new references returned by the server, where possible. This response is cacheable unless indicated otherwise.

The new permanent URI SHOULD be given by the Location field in the response. Unless the request method was HEAD, the entity of the response SHOULD contain a short hypertext note with a hyperlink to the new URI(s).

If the 301 status code is received in response to a request other than GET or HEAD, the user agent MUST NOT automatically redirect the request unless it can be confirmed by the user, since this might change the conditions under which the request was issued.

Note: When automatically redirecting a POST request after

receiving a 301 status code, some existing HTTP/1.0 user agents

will erroneously change it into a GET request.

10.3.3 302 Found

The requested resource resides temporarily under a different URI. Since the redirection might be altered on occasion, the client SHOULD continue to use the Request-URI for future requests. This response is only cacheable if indicated by a Cache-Control or Expires header field.

The temporary URI SHOULD be given by the Location field in the response. Unless the request method was HEAD, the entity of the response SHOULD contain a short hypertext note with a hyperlink to the new URI(s).

If the 302 status code is received in response to a request other than GET or HEAD, the user agent MUST NOT automatically redirect the request unless it can be confirmed by the user, since this might change the conditions under which the request was issued.

Note: RFC 1945 and RFC 2068 specify that the client is not allowed

to change the method on the redirected request. However, most

existing user agent implementations treat 302 as if it were a 303

response, performing a GET on the Location field-value regardless

of the original request method. The status codes 303 and 307 have

been added for servers that wish to make unambiguously clear which

kind of reaction is expected of the client.

10.3.4 303 See Other

The response to the request can be found under a different URI and SHOULD be retrieved using a GET method on that resource. This method exists primarily to allow the output of a POST-activated script to redirect the user agent to a selected resource. The new URI is not a substitute reference for the originally requested resource. The 303 response MUST NOT be cached, but the response to the second (redirected) request might be cacheable.

The different URI SHOULD be given by the Location field in the response. Unless the request method was HEAD, the entity of the response SHOULD contain a short hypertext note with a hyperlink to the new URI(s).

Note: Many pre-HTTP/1.1 user agents do not understand the 303

status. When interoperability with such clients is a concern, the

302 status code may be used instead, since most user agents react

to a 302 response as described here for 303.

10.3.5 304 Not Modified

If the client has performed a conditional GET request and access is allowed, but the document has not been modified, the server SHOULD respond with this status code. The 304 response MUST NOT contain a message-body, and thus is always terminated by the first empty line after the header fields.

The response MUST include the following header fields:

- Date, unless its omission is required by section 14.18.1

If a clockless origin server obeys these rules, and proxies and clients add their own Date to any response received without one (as already specified by [RFC 2068], section [14.19](https://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.19)), caches will operate correctly.

- ETag and/or Content-Location, if the header would have been sent

in a 200 response to the same request

- Expires, Cache-Control, and/or Vary, if the field-value might

differ from that sent in any previous response for the same

variant

If the conditional GET used a strong cache validator (see section 13.3.3), the response SHOULD NOT include other entity-headers. Otherwise (i.e., the conditional GET used a weak validator), the response MUST NOT include other entity-headers; this prevents inconsistencies between cached entity-bodies and updated headers.

If a 304 response indicates an entity not currently cached, then the cache MUST disregard the response and repeat the request without the conditional.

If a cache uses a received 304 response to update a cache entry, the cache MUST update the entry to reflect any new field values given in the response.

10.3.6 305 Use Proxy

The requested resource MUST be accessed through the proxy given by the Location field. The Location field gives the URI of the proxy. The recipient is expected to repeat this single request via the proxy. 305 responses MUST only be generated by origin servers.

Note: RFC 2068 was not clear that 305 was intended to redirect a

single request, and to be generated by origin servers only. Not

observing these limitations has significant security consequences.

10.3.7 306 (Unused)

The 306 status code was used in a previous version of the specification, is no longer used, and the code is reserved.

10.3.8 307 Temporary Redirect

The requested resource resides temporarily under a different URI. Since the redirection MAY be altered on occasion, the client SHOULD continue to use the Request-URI for future requests. This response is only cacheable if indicated by a Cache-Control or Expires header field.

The temporary URI SHOULD be given by the Location field in the response. Unless the request method was HEAD, the entity of the response SHOULD contain a short hypertext note with a hyperlink to the new URI(s) , since many pre-HTTP/1.1 user agents do not understand the 307 status. Therefore, the note SHOULD contain the information necessary for a user to repeat the original request on the new URI.

If the 307 status code is received in response to a request other than GET or HEAD, the user agent MUST NOT automatically redirect the request unless it can be confirmed by the user, since this might change the conditions under which the request was issued.

10.4 Client Error 4xx

The 4xx class of status code is intended for cases in which the client seems to have erred. Except when responding to a HEAD request, the server SHOULD include an entity containing an explanation of the error situation, and whether it is a temporary or permanent condition. These status codes are applicable to any request method. User agents SHOULD display any included entity to the user.

If the client is sending data, a server implementation using TCP SHOULD be careful to ensure that the client acknowledges receipt of the packet(s) containing the response, before the server closes the input connection. If the client continues sending data to the server after the close, the server's TCP stack will send a reset packet to the client, which may erase the client's unacknowledged input buffers before they can be read and interpreted by the HTTP application.

10.4.1 400 Bad Request

The request could not be understood by the server due to malformed syntax. The client SHOULD NOT repeat the request without modifications.

10.4.2 401 Unauthorized

The request requires user authentication. The response MUST include a WWW-Authenticate header field (section 14.47) containing a challenge applicable to the requested resource. The client MAY repeat the request with a suitable Authorization header field (section [14.8](https://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.8)). If the request already included Authorization credentials, then the 401 response indicates that authorization has been refused for those credentials. If the 401 response contains the same challenge as the prior response, and the user agent has already attempted authentication at least once, then the user SHOULD be presented the entity that was given in the response, since that entity might include relevant diagnostic information. HTTP access authentication is explained in "HTTP Authentication: Basic and Digest Access Authentication" [[43]](https://www.w3.org/Protocols/rfc2616/rfc2616-sec17.html#bib43).

10.4.3 402 Payment Required

This code is reserved for future use.

10.4.4 403 Forbidden

The server understood the request, but is refusing to fulfill it. Authorization will not help and the request SHOULD NOT be repeated. If the request method was not HEAD and the server wishes to make public why the request has not been fulfilled, it SHOULD describe the reason for the refusal in the entity. If the server does not wish to make this information available to the client, the status code 404 (Not Found) can be used instead.

10.4.5 404 Not Found

The server has not found anything matching the Request-URI. No indication is given of whether the condition is temporary or permanent. The 410 (Gone) status code SHOULD be used if the server knows, through some internally configurable mechanism, that an old resource is permanently unavailable and has no forwarding address. This status code is commonly used when the server does not wish to reveal exactly why the request has been refused, or when no other response is applicable.

10.4.6 405 Method Not Allowed

The method specified in the Request-Line is not allowed for the resource identified by the Request-URI. The response MUST include an Allow header containing a list of valid methods for the requested resource.

10.4.7 406 Not Acceptable

The resource identified by the request is only capable of generating response entities which have content characteristics not acceptable according to the accept headers sent in the request.

Unless it was a HEAD request, the response SHOULD include an entity containing a list of available entity characteristics and location(s) from which the user or user agent can choose the one most appropriate. The entity format is specified by the media type given in the Content-Type header field. Depending upon the format and the capabilities of the user agent, selection of the most appropriate choice MAY be performed automatically. However, this specification does not define any standard for such automatic selection.

Note: HTTP/1.1 servers are allowed to return responses which are

not acceptable according to the accept headers sent in the

request. In some cases, this may even be preferable to sending a

406 response. User agents are encouraged to inspect the headers of

an incoming response to determine if it is acceptable.

If the response could be unacceptable, a user agent SHOULD temporarily stop receipt of more data and query the user for a decision on further actions.

10.4.8 407 Proxy Authentication Required

This code is similar to 401 (Unauthorized), but indicates that the client must first authenticate itself with the proxy. The proxy MUST return a Proxy-Authenticate header field (section [14.33](https://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.33)) containing a challenge applicable to the proxy for the requested resource. The client MAY repeat the request with a suitable Proxy-Authorization header field (section [14.34](https://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.34)). HTTP access authentication is explained in "HTTP Authentication: Basic and Digest Access Authentication" [[43]](https://www.w3.org/Protocols/rfc2616/rfc2616-sec17.html#bib43).

10.4.9 408 Request Timeout

The client did not produce a request within the time that the server was prepared to wait. The client MAY repeat the request without modifications at any later time.

10.4.10 409 Conflict

The request could not be completed due to a conflict with the current state of the resource. This code is only allowed in situations where it is expected that the user might be able to resolve the conflict and resubmit the request. The response body SHOULD include enough

information for the user to recognize the source of the conflict. Ideally, the response entity would include enough information for the user or user agent to fix the problem; however, that might not be possible and is not required.

Conflicts are most likely to occur in response to a PUT request. For example, if versioning were being used and the entity being PUT included changes to a resource which conflict with those made by an earlier (third-party) request, the server might use the 409 response to indicate that it can't complete the request. In this case, the response entity would likely contain a list of the differences between the two versions in a format defined by the response Content-Type.

10.4.11 410 Gone

The requested resource is no longer available at the server and no forwarding address is known. This condition is expected to be considered permanent. Clients with link editing capabilities SHOULD delete references to the Request-URI after user approval. If the server does not know, or has no facility to determine, whether or not the condition is permanent, the status code 404 (Not Found) SHOULD be used instead. This response is cacheable unless indicated otherwise.

The 410 response is primarily intended to assist the task of web maintenance by notifying the recipient that the resource is intentionally unavailable and that the server owners desire that remote links to that resource be removed. Such an event is common for limited-time, promotional services and for resources belonging to individuals no longer working at the server's site. It is not necessary to mark all permanently unavailable resources as "gone" or to keep the mark for any length of time -- that is left to the discretion of the server owner.

10.4.12 411 Length Required

The server refuses to accept the request without a defined Content- Length. The client MAY repeat the request if it adds a valid Content-Length header field containing the length of the message-body in the request message.

10.4.13 412 Precondition Failed

The precondition given in one or more of the request-header fields evaluated to false when it was tested on the server. This response code allows the client to place preconditions on the current resource metainformation (header field data) and thus prevent the requested method from being applied to a resource other than the one intended.

10.4.14 413 Request Entity Too Large

The server is refusing to process a request because the request entity is larger than the server is willing or able to process. The server MAY close the connection to prevent the client from continuing the request.

If the condition is temporary, the server SHOULD include a Retry- After header field to indicate that it is temporary and after what time the client MAY try again.

10.4.15 414 Request-URI Too Long

The server is refusing to service the request because the Request-URI is longer than the server is willing to interpret. This rare condition is only likely to occur when a client has improperly converted a POST request to a GET request with long query information, when the client has descended into a URI "black hole" of redirection (e.g., a redirected URI prefix that points to a suffix of itself), or when the server is under attack by a client attempting to exploit security holes present in some servers using fixed-length buffers for reading or manipulating the Request-URI.

10.4.16 415 Unsupported Media Type

The server is refusing to service the request because the entity of the request is in a format not supported by the requested resource for the requested method.

10.4.17 416 Requested Range Not Satisfiable

A server SHOULD return a response with this status code if a request included a Range request-header field (section 14.35), and none of the range-specifier values in this field overlap the current extent of the selected resource, and the request did not include an If-Range request-header field. (For byte-ranges, this means that the first- byte-pos of all of the byte-range-spec values were greater than the current length of the selected resource.)

When this status code is returned for a byte-range request, the response SHOULD include a Content-Range entity-header field specifying the current length of the selected resource (see section [14.16](https://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.16)). This response MUST NOT use the multipart/byteranges content- type.

10.4.18 417 Expectation Failed

The expectation given in an Expect request-header field (see section 14.20) could not be met by this server, or, if the server is a proxy, the server has unambiguous evidence that the request could not be met by the next-hop server.

10.5 Server Error 5xx

Response status codes beginning with the digit "5" indicate cases in which the server is aware that it has erred or is incapable of performing the request. Except when responding to a HEAD request, the server SHOULD include an entity containing an explanation of the error situation, and whether it is a temporary or permanent condition. User agents SHOULD display any included entity to the user. These response codes are applicable to any request method.

10.5.1 500 Internal Server Error

The server encountered an unexpected condition which prevented it from fulfilling the request.

10.5.2 501 Not Implemented

The server does not support the functionality required to fulfill the request. This is the appropriate response when the server does not recognize the request method and is not capable of supporting it for any resource.

10.5.3 502 Bad Gateway

The server, while acting as a gateway or proxy, received an invalid response from the upstream server it accessed in attempting to fulfill the request.

10.5.4 503 Service Unavailable

The server is currently unable to handle the request due to a temporary overloading or maintenance of the server. The implication is that this is a temporary condition which will be alleviated after some delay. If known, the length of the delay MAY be indicated in a Retry-After header. If no Retry-After is given, the client SHOULD handle the response as it would for a 500 response.

Note: The existence of the 503 status code does not imply that a

server must use it when becoming overloaded. Some servers may wish

to simply refuse the connection.

10.5.5 504 Gateway Timeout

The server, while acting as a gateway or proxy, did not receive a timely response from the upstream server specified by the URI (e.g. HTTP, FTP, LDAP) or some other auxiliary server (e.g. DNS) it needed to access in attempting to complete the request.

Note: Note to implementors: some deployed proxies are known to

return 400 or 500 when DNS lookups time out.

10.5.6 505 HTTP Version Not Supported

The server does not support, or refuses to support, the HTTP protocol version that was used in the request message. The server is indicating that it is unable or unwilling to complete the request using the same major version as the client, as described in section [3.1](https://www.w3.org/Protocols/rfc2616/rfc2616-sec3.html#sec3.1), other than with this error message. The response SHOULD contain an entity describing why that version is not supported and what other protocols are supported by that server.