

WIKIPEDIA

Anexo:Códigos de estado HTTP

La siguiente es una lista de códigos de respuesta del HTTP y, separadas por un guion, la frase estándar asociadas a ese código de respuesta en el protocolo HTTP. Estas frases estándares destinadas a dar una descripción intuitiva del estatus. Los códigos de estatus están especificados por el [RFC 2616](#) (actualmente obsoleto), y algunos fragmentos en los estándares [RFC 2518](#) (también obsoleto), [RFC 2817](#), [RFC 2295](#) (experimental), [RFC 2774](#) (también experimental) y [RFC 4918](#); otros no están estandarizados, pero son comúnmente utilizados.

El primer dígito del código de respuesta especifica una de las cinco clases de respuesta.

Índice

1xx: Respuestas informativas

2xx: Peticiones correctas

3xx: Redirecciones

4xx Errores del cliente

5xx Errores de servidor

Véase también

Referencias

Enlaces externos

1xx: Respuestas informativas

Petición recibida, continuando proceso. Esta respuesta significa que el servidor ha recibido los encabezados de la petición, y que el cliente debería proceder a enviar el cuerpo de la misma (en el caso de peticiones para las cuales el cuerpo necesita ser enviado; por ejemplo, una petición [Hypertext Transfer Protocol](#)). Si el cuerpo de la petición es largo, es ineficiente enviarlo a un servidor, cuando la petición ha sido ya rechazada, debido a encabezados inapropiados. Para hacer que un servidor chequee si la petición podría ser aceptada basada únicamente en los encabezados de la petición, el cliente debe enviar `Expect: 100-continue` como un encabezado en su petición inicial (vea [Plantilla:Web-RFC: Expect header](#)) y verificar si un código de estado `100 Continue` es recibido en respuesta, antes de continuar (o recibir `417 Expectation Failed` y no continuar).¹

100 Continue

El navegador puede continuar realizando su petición (se utiliza para indicar que la primera parte de la petición del navegador se ha recibido correctamente).²

101 Switching Protocols

El servidor acepta el cambio de protocolo propuesto por el navegador (puede ser por ejemplo un cambio de HTTP 1.0 a HTTP 1.1).²

102 Processing (WebDAV - RFC 2518)

El servidor está procesando la petición del navegador pero todavía no ha terminado (esto evita que el navegador piense que la petición se ha perdido cuando no recibe ninguna respuesta).²

103 Checkpoint

Se va a reanudar una petición POST o PUT que fue abortada previamente.²

2xx: Peticiones correctas

Esta clase de código de estado indica que la petición fue recibida correctamente, entendida y aceptada.

200 OK

Respuesta estándar para peticiones correctas.

201 Created

La petición ha sido completada y ha resultado en la creación de un nuevo recurso.

202 Accepted

La petición ha sido aceptada para procesamiento, pero este no ha sido completado. La petición eventualmente pudiere no ser satisfecha, ya que podría ser no permitida o prohibida cuando el procesamiento tenga lugar.

203 Non-Authoritative Information (desde HTTP/1.1)

La petición se ha completado con éxito, pero su contenido no se ha obtenido de la fuente originalmente solicitada sino de otro servidor.²

204 No Content

La petición se ha completado con éxito pero su respuesta no tiene ningún contenido (la respuesta sí que puede incluir información en sus cabeceras HTTP).²

205 Reset Content

La petición se ha completado con éxito, pero su respuesta no tiene contenidos y además, el navegador tiene que inicializar la página desde la que se realizó la petición (este código es útil por ejemplo para páginas con formularios cuyo contenido debe borrarse después de que el usuario lo envíe).²

206 Partial Content

La petición servirá parcialmente el contenido solicitado. Esta característica es utilizada por herramientas de descarga como `wget` para continuar la transferencia de descargas anteriormente interrumpidas, o para dividir una descarga y procesar las partes simultáneamente.

207 Multi-Status (Multi-Status, WebDAV)

El cuerpo del mensaje que sigue es un mensaje XML y puede contener algún número de códigos de respuesta separados, dependiendo de cuántas sub-peticiones sean hechas.

208 Already Reported (WebDAV)

El listado de elementos DAV ya se notificó previamente, por lo que no se van a volver a listar.²

3xx: Redirecciones

El cliente tiene que tomar una acción adicional para completar la petición.

Esta clase de código de estado indica que una acción subsecuente necesita efectuarse por el agente de usuario para completar la petición. La acción requerida *puede* ser llevada a cabo por el **agente de usuario** sin interacción con el usuario si y solo si el método utilizado en la segunda petición es GET o HEAD. El agente de usuario *no debe* redirigir automáticamente una petición más de 5 veces, dado que tal funcionamiento indica usualmente un Bucle infinito.

300 Multiple Choices

Indica opciones múltiples para el URI que el cliente podría seguir. Esto podría ser utilizado, por ejemplo, para presentar distintas opciones de formato para video, listar archivos con distintas extensiones o word sense disambiguation.

301 Moved Permanently

Esta y todas las peticiones futuras deberían ser dirigidas a la URI dada.

302 Found

Este es el código de redirección más popular, pero también un ejemplo de las prácticas de la industria contradiciendo el estándar. La especificación HTTP/1.0 (RFC 1945) requería que el cliente realizara una redirección temporal (la frase descriptiva original fue "Moved Temporarily"), pero los navegadores populares lo implementaron como **303 See Other**. Por tanto, HTTP/1.1 añadió códigos de estado 303 y 307 para eliminar la ambigüedad entre ambos comportamientos. Sin embargo, la mayoría de aplicaciones web y bibliotecas de desarrollo aún utilizan el código de respuesta 302 como si fuera el 303.

303 See Other (desde HTTP/1.1)

La respuesta a la petición puede ser encontrada bajo otra URI utilizando el método GET.

304 Not Modified

Indica que la petición a la URL no ha sido modificada desde que fue requerida por última vez. Típicamente, el cliente HTTP provee un encabezado como `If-Modified-Since` para indicar una fecha y hora contra la cual el servidor pueda comparar. El uso de este encabezado ahorra ancho de banda y reprocesamiento tanto del servidor como del cliente.

305 Use Proxy (desde HTTP/1.1)

Muchos clientes HTTP (como Mozilla³ e Internet Explorer) no se apegan al estándar al procesar respuestas con este código, principalmente por motivos de seguridad.

306 Switch Proxy

Este código se utilizaba en las versiones antiguas de HTTP pero ya no se usa (aunque está reservado para usos futuros).²

307 Temporary Redirect (desde HTTP/1.1)

Se trata de una redirección que debería haber sido hecha con otra URI, sin embargo aún puede ser procesada con la URI proporcionada. En contraste con el código 303, el método de la petición no debería ser cambiado cuando el cliente repita la solicitud. Por ejemplo, una solicitud POST tiene que ser repetida utilizando otra petición POST.

308 Permanent Redirect

El recurso solicitado por el navegador se encuentra en otro lugar y este cambio es permanente. A diferencia del código 301, no se permite cambiar el método HTTP para la nueva petición (así por ejemplo, si envías un formulario a un recurso que ha cambiado de lugar, todo seguirá funcionando bien).²

4xx Errores del cliente

La solicitud contiene sintaxis incorrecta o no puede procesarse.

La intención de la clase de códigos de respuesta 4xx es para casos en los cuales el cliente parece haber errado la petición. Excepto cuando se responde a una petición HEAD, el servidor **debe** incluir una entidad que contenga una explicación a la situación de error, y si es una condición temporal o permanente. Estos códigos de estado son aplicables a cualquier método de solicitud (como GET o POST). Los agentes de usuario *deben* desplegar cualquier entidad al usuario. Estos son típicamente los códigos de respuesta de error más comúnmente encontrados.

400 Bad Request

La solicitud contiene sintaxis errónea y no debería repetirse.

401 Unauthorized

Similar al *403 Forbidden*, pero específicamente para su uso cuando la autenticación es posible pero ha fallado o aún no ha sido provista. Vea autenticación HTTP básica y Digest access authentication.

402 Payment Required

La intención original era que este código pudiese ser usado como parte de alguna forma o esquema de Dinero electrónico o micropagos, pero eso no sucedió, y este código nunca se utilizó.

403 Forbidden

La solicitud fue legal, pero el servidor rehúsa responderla dado que el cliente no tiene los privilegios para hacerla. En contraste a una respuesta *401 No autorizado*, la autenticación no haría la diferencia.

404 Not Found

Recurso no encontrado. Se utiliza cuando el servidor web no encuentra la página o recurso solicitado.

405 Method Not Allowed

Una petición fue hecha a una URI utilizando un método de solicitud no soportado por dicha URI; por ejemplo, cuando se utiliza GET en un formulario que requiere que los datos sean presentados vía POST, o utilizando PUT en un recurso de solo lectura.

406 Not Acceptable

El servidor no es capaz de devolver los datos en ninguno de los formatos aceptados por el cliente, indicados por éste en la cabecera "Accept" de la petición.

407 Proxy Authentication Required

408 Request Timeout

El cliente falló al continuar la petición - excepto durante la ejecución de videos Adobe Flash cuando solo significa que el usuario cerró la ventana de video o se movió a otro. ref (<http://livedocs.adobe.com/fms/2/docs/00000338.html>)

409 Conflict

Indica que la solicitud no pudo ser procesada debido a un conflicto con el estado actual del recurso que esta identifica.

410 Gone

Indica que el recurso solicitado ya no está disponible y no lo estará de nuevo. Debería ser utilizado cuando un recurso ha sido quitado de forma permanente.

Si un cliente recibe este código no debería volver a solicitar el recurso en el futuro. Por ejemplo un buscador lo eliminará de sus índices y lo hará más rápidamente que utilizando un código 404.

411 Length Required

El servidor rechaza la petición del navegador porque no incluye la cabecera Content-Length adecuada.²

412 Precondition Failed

El servidor no es capaz de cumplir con algunas de las condiciones impuestas por el navegador en su petición.²

413 Request Entity Too Large

La petición del navegador es demasiado grande y por ese motivo el servidor no la procesa²

414 Request-URI Too Long

La URI de la petición del navegador es demasiado grande y por ese motivo el servidor no la procesa (esta condición se produce en muy raras ocasiones y casi siempre porque el navegador envía como GET una petición que debería ser POST).²

415 Unsupported Media Type

La petición del navegador tiene un formato que no entiende el servidor y por eso no se procesa.²

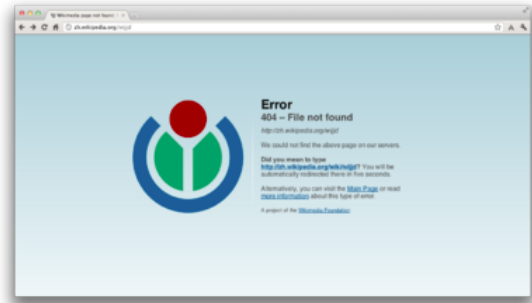
416 Requested Range Not Satisfiable

El cliente ha preguntado por una parte de un archivo, pero el servidor no puede proporcionar esa parte, por ejemplo, si el cliente preguntó por una parte de un archivo que está más allá de los límites del fin del archivo.

417 Expectation Failed

La petición del navegador no se procesa porque el servidor no es capaz de cumplir con los requerimientos de la cabecera Expect de la petición.²

418 I'm a teapot



El error 404 en Wikipedia

"Soy una tetera". Este código fue definido en 1998 como una inocentada, en el Protocolo de Transmisión de Hipertexto de Cafeteras (RFC-2324 (<https://www.ietf.org/rfc/rfc2324.txt>)). No se espera que los servidores web implementen realmente este código de error, pero es posible encontrar sitios que devuelvan este código HTTP.

422 Unprocessable Entity (WebDAV - RFC 4918)

La solicitud está bien formada pero fue imposible seguirla debido a errores semánticos.

423 Locked (WebDAV - RFC 4918)

El recurso al que se está teniendo acceso está bloqueado.

424 Failed Dependency (WebDAV) (RFC 4918)

La solicitud falló debido a una falla en la solicitud previa.

425 Unassigned

Definido en los **drafts** de WebDav Advanced Collections (<http://tools.ietf.org/html/draft-ietf-webdav-collection-protocol-04#section-5.3.2>), pero no está presente en "Web Distributed Authoring and Versioning (WebDAV) Ordered Collections Protocol" (RFC 3648).

426 Upgrade Required (RFC 7231)

El cliente debería cambiarse a [TLS/1.0](#).

428 Precondition Required

El servidor requiere que la petición del navegador sea condicional (este tipo de peticiones evitan los problemas producidos al modificar con PUT un recurso que ha sido modificado por otra parte).²

429 Too Many Requests

Hay muchas conexiones desde esta dirección de internet.

431 Request Header Fields Too Large

El servidor no puede procesar la petición porque una de las cabeceras de la petición es demasiado grande. Este error también se produce cuando la suma del tamaño de todas las peticiones es demasiado grande.²

449

Una extensión de Microsoft: *La petición debería ser reintentada después de hacer la acción apropiada*.

451 Unavailable for Legal Reasons

El contenido ha sido eliminado como consecuencia de una orden judicial o sentencia emitida por un tribunal.

5xx Errores de servidor

El servidor falló al completar una solicitud aparentemente válida.

Los códigos de respuesta que comienzan con el dígito "5" indican casos en los cuales el servidor tiene registrado aún antes de servir la solicitud, que está errado o es incapaz de ejecutar la petición. Excepto cuando está respondiendo a un método HEAD, el servidor *debe* incluir una entidad que contenga una explicación de la situación de error, y si es una condición temporal o permanente. Los agentes de usuario *deben* desplegar cualquier entidad incluida al usuario. Estos códigos de respuesta son aplicables a cualquier método de petición.

500 Internal Server Error

Es un código comúnmente emitido por aplicaciones empotradas en servidores web, mismas que generan contenido dinámicamente, por ejemplo aplicaciones montadas en [IIS](#) o [Tomcat](#), cuando se encuentran con situaciones de error ajenas a la naturaleza del servidor web.

501 Not Implemented

El servidor no soporta alguna funcionalidad necesaria para responder a la solicitud del navegador (como por ejemplo el método utilizado para la petición).²

502 Bad Gateway

El servidor está actuando de proxy o gateway y ha recibido una respuesta inválida del otro servidor, por lo que no puede responder adecuadamente a la petición del navegador.²

503 Service Unavailable

El servidor no puede responder a la petición del navegador porque está congestionado o está realizando tareas de mantenimiento.²

504 Gateway Timeout

El servidor está actuando de proxy o gateway y no ha recibido a tiempo una respuesta del otro servidor, por lo que no puede responder adecuadamente a la petición del navegador.²

505 HTTP Version Not Supported

El servidor no soporta o no quiere soportar la versión del protocolo HTTP utilizada en la petición del navegador.²

506 Variant Also Negotiates (RFC 2295)

El servidor ha detectado una referencia circular al procesar la parte de la negociación del contenido de la petición.²

507 Insufficient Storage (WebDAV - RFC 4918)

El servidor no puede crear o modificar el recurso solicitado porque no hay suficiente espacio de almacenamiento libre.²

508 Loop Detected (WebDAV)

La petición no se puede procesar porque el servidor ha encontrado un bucle infinito al intentar procesarla.²

509 Bandwidth Limit Exceeded

Límite de ancho de banda excedido. Este código de estatus, a pesar de ser utilizado por muchos servidores, no

es oficial.

510 Not Extended (RFC 2774)

La petición del navegador debe añadir más extensiones para que el servidor pueda procesarla.²

511 Network Authentication Required

El navegador debe autenticarse para poder realizar peticiones (se utiliza por ejemplo con los portales cautivos que te obligan a autenticarte antes de empezar a navegar).²

512 Not updated

Este error prácticamente es inexistente en la red, pero indica que el servidor está en una operación de actualizado y no puede tener conexión.

Véase también

- [Hypertext Transfer Protocol](#)

Referencias

- ↑ RFC 2616, section 8.2.3 (<https://tools.ietf.org/html/rfc2616#section-8.2.3>) — Use of the 100 (Continue) Status
- ↑ Los códigos de estado de HTTP - LibrosWeb (<http://librosweb.es/tutorial/los-codigos-de-estado-de-http/>)
- ↑ Bugzilla (https://bugzilla.mozilla.org/show_bug.cgi?id=187996)

Enlaces externos

- HTTP specification, Section 10 (<https://tools.ietf.org/html/rfc2616#section-10>)
 - WebDAV specification, Section 11 (<https://tools.ietf.org/html/rfc4918#section-11>)
 - TLS Upgrade within HTTP specification, Section 4 (<https://tools.ietf.org/html/rfc2817#section-4>)
 - HTTP Status Code Registry (<https://tools.ietf.org/html/rfc2817#section-7.1>)
 - IANA registry (<http://www.iana.org/assignments/http-status-codes>)
 - Microsoft Internet Information Server Status Codes and Sub-Codes (<http://support.microsoft.com/?id=318380>)
 - **Adobe Flash status code definitions (ie 408)** (<http://livedocs.adobe.com/fms/2/docs/00000338.html>)
-

Obtenido de «https://es.wikipedia.org/w/index.php?title=Anexo:Códigos_de_estado_HTTP&oldid=107008397»

Se editó esta página por última vez el 13 abr 2018 a las 15:57.

El texto está disponible bajo la [Licencia Creative Commons Atribución Compartir Igual 3.0](#); pueden aplicarse cláusulas adicionales. Al usar este sitio, usted acepta nuestros [términos de uso](#) y nuestra [política de privacidad](#).
Wikipedia® es una marca registrada de la [Fundación Wikimedia, Inc.](#), una organización sin ánimo de lucro.