

Tipo de respuestas

1. 1XX Respuestas informativas
2. 2XX Peticiones correctas
3. 3XX Redirecciones
4. 4XX Errores del cliente
5. 5XX Errores de servidor

1. 1XX Respuestas informativas

Este tipo de código de estado indica una **respuesta provisional**.

HTTP/1.0 no definió ningún estado 1XX, por lo que los servidores no deberían enviar una respuesta 1XX a un cliente HTTP/1.0 salvo para experimentos.

- **100 Continue.** El servidor ha recibido los **headers del request** y el cliente debería proceder a enviar el cuerpo de la respuesta.
- **101 Switching Protocols.** El **requester** ha solicitado al servidor conmutar protocolos.
- **102 Processing (WebDAV; RFC 2518).** Usado en **requests** para reanudar peticiones PUT o POST abortadas.

2. 2XX Peticiones correctas

Este código de estado indica que la acción solicitada por el **cliente** ha sido recibida, entendida, aceptada y procesada correctamente.

- **200 OK.** El request es correcto. Esta es la respuesta estándar para respuestas correctas.
- **201 Created.** El request se ha completado y se ha creado un nuevo recurso.
- **202 Aceptada.** El request se ha aceptado para procesarlo, pero el proceso aún no ha terminado.

- **203 Non-Authoritative Information.** El request se ha procesado correctamente, pero devuelve información que podría venir de otra fuente.
- **204 No Content.** El request se ha procesado correctamente, pero no devuelve ningún contenido.
- **205 Reset Content.** El request se ha procesado correctamente, pero no devuelve ningún contenido y se requiere que el requester recargue el contenido.
- **206 Partial Content.** El servidor devuelve sólo parte del recurso debido a una limitación que ha configurado el cliente (se usa en herramientas de descarga como wget).
- **207 Multi-Status (WebDAV; RFC 4918).** El cuerpo del mensaje es XML y puede contener un número de códigos de estado diferentes dependiendo del número de sub-requests.

3. 3XX Redirecciones

El cliente ha de tomar una acción adicional para completar el **request**. Muchos de estos estados se utilizan para **redirecciones**. El **user-agent** puede llevar a cabo la acción adicional sin necesidad de que actúe el **usuario** sólo si el **método** utilizado en la segunda petición es **GET** o **HEAD**. Un user-agent no debería redireccionar automáticamente un request más de cinco veces, ya que esas redirecciones suelen indicar un **infinite loop**.

- **300 Multiple Choices.** Es una lista de enlaces. El usuario puede seleccionar un enlace e ir a esa dirección. Hay un máximo de cinco direcciones.
- **301 Moved Permanently.** La página solicitada se ha movido permanentemente a una nueva URI.
- **302 Found.** La página solicitada se ha movido temporalmente a una nueva URI.

- **303 See Other.** La página solicitada se puede encontrar en una URI diferente.
- **304 Not Modified.** Indica que la página solicitada no se ha modificado desde la última petición.
- **305 Use Proxy (desde HTTP/1.1).** El recurso solicitado sólo está disponible a través de proxy, cuya dirección se proporciona en la respuesta. Muchos clientes HTTP como Mozilla o Internet Explorer no manejan bien estas respuestas con estos códigos de estado, sobre todo por seguridad.
- **307 Temporary Redirect (desde HTTP/1.1).** La página solicitada se ha movido temporalmente a otra URL. En este caso el recurso debería de repetirse con otra URI, sin embargo, futuros requests deberán usar la URI original. Al contrario que con la 302, el método request no puede cambiar cuando se reubique el request original.
- **308 Permanent Redirect (RFC 7538).** El request y futuros requests deberían repetirse usando otro URI. Este también es similar al 301, pero no permite al método HTTP que cambie.

4. 4XX Errores del cliente

Excepto cuando se responde a un **HEAD request**, el servidor debe incluir una entidad que contiene una explicación del error, y si es temporal o permanente. Son aplicables a cualquier método de solicitud (GET, POST...). Los **user agents** deben mostrar cualquier entidad al usuario:

- **400 Bad Request.** El servidor no puede o no va a procesar el request por un error de sintaxis del cliente.
- **401 Unauthorized (RFC 7235).** Similar al error 403, pero se usa cuando se requiere una autenticación y ha fallado o todavía no se a facilitado.

- **402 Payment Required.** Reservado para futuro uso. La intención original fue para **pago con tarjeta o micropago**, pero eso no ha ocurrido, y este código apenas se usa.
- **403 Forbidden.** El request fue válido pero **el servidor se niega a responder**.
- **404 Not Found.** El recurso del request no se ha podido encontrar pero podría estar disponible en el futuro. Se permiten **requests subsecuentes** por parte del cliente.
- **405 Method Not Allowed.** Se ha hecho un request con un recurso usando un **método request no soportado** por ese recurso (por ejemplo usando GET en un formulario que requiere POST).
- **406 Not Acceptable.** El recurso solicitado solo genera **contenido no aceptado** de acuerdo con los headers Accept enviados en el request.
- **407 Proxy Authentication Required (RFC 7235).** El cliente se debe identificar primero con el **proxy**.
- **408 Request Timeout.** El cliente no ha enviado un **request con el tiempo necesario** con el que el servidor estaba preparado para esperar. El cliente podría repetir el request sin modificaciones más tarde.
- **409 Conflict.** Conflicto en el request, como cuando se actualizan al mismo tiempo dos recursos.
- **410 Gone.** El recurso solicitado no está disponible ni lo estará en el futuro. **Un buscador eliminará antes una página 410 que una 404.**
- **411 Length Required.** El request no especificó la **longitud del contenido**, la cual es requerida por el recurso solicitado.
- **412 Precondition Failed (RFC 7232).** El servidor no cumple una de las precondiciones que el requester añade en el request.

- **413 Request Entity Too Large.** El request es más largo que el que está dispuesto a aceptar el servidor.
- **414 Request-URI Too Long.** El **URI** es muy largo para que el servidor lo procese.
- **415 Unsupported Media Type.** La entidad request tiene un **media type** que el servidor o recurso no soportan.
- **416 Requested Range Not Satisfiable (RFC 7233).** El cliente ha solicitado una porción de archivo, pero el servidor no puede ofrecer esa porción.
- **417 Expectation Failed.** El servidor no puede cumplir los requerimientos del header del request Expect.
- **418 I'm a teapot (RFC 2324).** Fue parte de un April Fool's day, y no se espera que se implemente en servidores HTTP. La RFC especifica que este código debería ser devuelto por teteras para servir té.

5. 5XX Errores del servidor

El servidor ha fallado al completar una solicitud aparentemente válida. Cuando los códigos de estado empiezan por 5 indica casos en los que **el servidor sabe que tiene un error** o realmente es **incapaz de procesar el request**. Salvo cuando se trata de un request **HEAD**, el servidor debe incluir una entidad conteniendo una **explicación del error**, y de si es temporal o permanente. Igualmente los user-agents deberán mostrar cualquier entidad al usuario. Estos códigos de respuesta se aplican a cualquier método request.

- **500 Internal Server Error.** Error genérico, cuando se ha dado una condición no esperada y no se puede concretar el mensaje.
- **501 Not Implemented.** El servidor o no reconoce el **método del request** o carece de la capacidad para completarlo. Normalmente

es algo que se ofrecerá en el futuro, como un **nuevo servicio** de una **API**.

- **502 Bad Gateway**. El server actuaba como puerta de entrada o **proxy** y recibió una respuesta inválida del servidor upstream.
- **503 Service Unavailable**. El servidor está actualmente no disponible, ya sea por **mantenimiento** o por **sobrecarga**.
- **504 Gateway Timeout**. El servidor estaba actuando como **puerta de entrada** o **proxy** y no recibió una respuesta oportuna por parte del servidor upstream.
- **505 HTTP Version Not Supported**. El servidor no soporta la **versión del protocolo HTTP** usada en el request.
- **511 Network Authentication Required (RFC 6585)**. El cliente necesita autenticarse para poder acceder a la red.