

Investigación

Métodos

GET

Esta petición retorna tanto la cabecera como el contenido. Ahora, este método GET puede retornar una respuesta en formato HTML, JSON, XML, Imágenes o JavaScript. Semánticamente se utiliza para consultar información como una SELECT a la base de datos, se puede filtrar datos empleando los datos enviados por la Url.

HEAD

Este método de petición es similar al método HTTP GET, sin embargo no retorna ningún contenido HTTP Response. Cuando se trabaja con este método de petición no se está interesado en el contenido, solo en el código de HTTP de Respuesta y el encabezado (Headers).

POST

Es otro de los métodos HTTP, este puede enviar datos al servidor por medio del cuerpo (body) y nada por la Url como se emplea en el método GET. El tipo de cuerpo de solicitud se define en la cabecera Content-Type. Semánticamente se utiliza para registrar información, similar al INSERT de datos a nivel de base de datos.

PUT

Es similar al método de petición POST, solo que el método PUT es idempotente; es decir puede ser ejecutado varias veces y tiene el mismo efecto, caso contrario a un POST que cada vez que se ejecuta realiza la agregación de un nuevo objeto, ya que semánticamente es como una inserción de un nuevo registro.

DELETE

Este método de petición permite eliminar un recurso específico. También es idempotente; es decir puede ser ejecutado varias veces y tiene el mismo efecto similar al PUT y GET.

PATCH Este método se emplea para modificaciones parciales de un recurso en particular. Se debe revisar si el servidor es compatible con peticiones PATCH. Para averiguar si el servidor acepta peticiones PATCH notifica su compatibilidad en el header Allow o Access-Control-Allow-Methods, otra indicación de que estas peticiones están permitidas son el header Accept-Patch.

OPTIONS

Este método se emplea para modificaciones parciales de un recurso en particular. Se debe revisar si el servidor es compatible con peticiones PATCH. Para averiguar si el servidor acepta peticiones PATCH notifica su compatibilidad en el header Allow o Access-Control-Allow-Methods, otra indicación de que estas peticiones están permitidas son el header Accept-Patch.

TRACE

Se trata de un método HTTP de petición muy interesante, ya que permite obtener la ruta que sigue una HTTP request durante todo su camino (desde que se realiza, hasta que llega al servidor y vuelve de regreso al cliente).

Códigos de respuesta

Respuestas informativas (100–199)

- 100 Continue

Esta respuesta provisional indica que todo hasta ahora está bien y que el cliente debe continuar con la solicitud o ignorarla si ya está terminada.

- 101 Switching Protocol Este código se envía en respuesta a un encabezado de solicitud Upgrade (en-US) por el cliente e indica que el servidor acepta el cambio de protocolo propuesto por el agente de usuario.

- 102 Processing (WebDAV (en-US)) Este código indica que el servidor ha recibido la solicitud y aún se encuentra procesandola, por lo que no hay respuesta disponible.

- 103 Early Hints (en-US) Este código de estado está pensado principalmente para ser usado con el encabezado Link, permitiendo que el agente de usuario empiece a pre-cargar recursos mientras el servidor prepara una respuesta.

Respuestas satisfactorias (200–299)

200 OK

La solicitud ha tenido éxito. El significado de un éxito varía dependiendo del método HTTP:

201 Created

La solicitud ha tenido éxito y se ha creado un nuevo recurso como resultado de ello. Ésta es típicamente la respuesta enviada después de una petición PUT.

202 Accepted

La solicitud se ha recibido, pero aún no se ha actuado. Es una petición "sin compromiso", lo que significa que no hay manera en HTTP que permite enviar una respuesta asíncrona que indique el resultado del procesamiento de la solicitud. Está pensado para los casos en que otro proceso o servidor maneja la solicitud, o para el procesamiento por lotes.

203 Non-Authoritative Information

La petición se ha completado con éxito, pero su contenido no se ha obtenido de la fuente originalmente solicitada, sino que se recoge de una copia local o de un tercero. Excepto esta condición, se debe preferir una respuesta de 200 OK en lugar de esta respuesta.

204 No Content (en-US)

La petición se ha completado con éxito pero su respuesta no tiene ningún contenido, aunque los encabezados pueden ser útiles. El agente de usuario puede actualizar sus encabezados en caché para este recurso con los nuevos valores.

205 Reset Content (en-US)

La petición se ha completado con éxito, pero su respuesta no tiene contenidos y además, el agente de usuario 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.

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 (WebDAV (en-US)) Una respuesta Multi-Estado transmite información sobre varios recursos en situaciones en las que varios códigos de estado podrían ser apropiados. El cuerpo de la petición es un mensaje XML.

208 Multi-Status (WebDAV (en-US))

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

226 IM Used (HTTP Delta encoding) El servidor ha cumplido una petición GET para el recurso y la respuesta es una representación del resultado de una o más manipulaciones de instancia aplicadas a la instancia actual.

Redirecciones (300–399)

300 Multiple Choice (en-US)

Esta solicitud tiene más de una posible respuesta. User-Agent o el usuario debe escoger uno de ellos. No hay forma estandarizada de seleccionar una de las respuestas.

301 Moved Permanently (en-US) Este código de respuesta significa que la URI del recurso solicitado ha sido cambiado. Probablemente una nueva URI sea devuelta en la respuesta.

302 Found Este código de respuesta significa que el recurso de la URI solicitada ha sido cambiado temporalmente. Nuevos cambios en la URI serán agregados en el futuro. Por lo tanto, la misma URI debe ser usada por el cliente en futuras solicitudes.

303 See Other (en-US) El servidor envía esta respuesta para dirigir al cliente a un nuevo recurso solicitado a otra dirección usando una petición GET.

304 Not Modified Esta es usada para propósitos de "caché". Le indica al cliente que la respuesta no ha sido modificada. Entonces, el cliente puede continuar usando la misma versión almacenada en su caché.

305 Use Proxy Deprecated Fue definida en una versión previa de la especificación del protocolo HTTP para indicar que una respuesta solicitada debe ser accedida desde un proxy. Ha quedado obsoleta debido a preocupaciones de seguridad correspondientes a la configuración de un proxy.

306 unused Este código de respuesta ya no es usado más. Actualmente se encuentra reservado. Fue usado en previas versiones de la especificación HTTP1.1.

307 Temporary Redirect (en-US) El servidor envía esta respuesta para dirigir al cliente a obtener el recurso solicitado a otra URI con el mismo método que se usó la petición anterior. Tiene la misma semántica que el código de respuesta HTTP 302 Found, con la excepción de que el agente usuario no debe cambiar el método HTTP usado: si un POST fue usado en la primera petición, otro POST debe ser usado en la segunda petición.

308 Permanent Redirect (en-US) Significa que el recurso ahora se encuentra permanentemente en otra URI, especificada por la respuesta de encabezado HTTP Location:. Tiene la misma semántica que el código de respuesta HTTP 301 Moved Permanently, con la excepción de que el agente usuario no debe cambiar el método HTTP usado: si un POST fue usado en la primera petición, otro POST debe ser usado en la segunda petición.

Errores de los clientes (400–499)

400 Bad Request

Esta respuesta significa que el servidor no pudo interpretar la solicitud dada una sintaxis inválida.

401 Unauthorized

Es necesario autenticar para obtener la respuesta solicitada. Esta es similar a 403, pero en este caso, la autenticación es posible.

402 Payment Required

Este código de respuesta está reservado para futuros usos. El objetivo inicial de crear este código fue para ser utilizado en sistemas digitales de pagos. Sin embargo, no está siendo usado actualmente.

403 Forbidden

El cliente no posee los permisos necesarios para cierto contenido, por lo que el servidor está rechazando otorgar una respuesta apropiada.

404 Not Found

El servidor no pudo encontrar el contenido solicitado. Este código de respuesta es uno de los más famosos dada su alta ocurrencia en la web.

405 Method Not Allowed (en-US)

El método solicitado es conocido por el servidor pero ha sido deshabilitado y no puede ser utilizado. Los dos métodos obligatorios, GET y HEAD, nunca deben ser deshabilitados y no deberían retornar este código de error.

406 Not Acceptable (en-US)

Esta respuesta es enviada cuando el servidor, después de aplicar una negociación de contenido servidor-impulsado, no encuentra ningún contenido seguido por la criteria dada por el usuario.

407 Proxy Authentication Required (en-US)

Esto es similar al código 401, pero la autenticación debe estar hecha a partir de un proxy.

408 Request Timeout

Esta respuesta es enviada en una conexión inactiva en algunos servidores, incluso sin alguna petición previa por el cliente. Significa que el servidor quiere desconectar esta conexión sin usar. Esta respuesta es muy usada desde algunos navegadores, como Chrome, Firefox 27+, o IE9, usa mecanismos de pre-conexión HTTP para acelerar la navegación. También hay que tener en cuenta que algunos servidores simplemente desconecta la conexión sin enviar este mensaje.

409 Conflict (en-US)

Esta respuesta puede ser enviada cuando una petición tiene conflicto con el estado actual del servidor.

410 Gone (en-US)

Esta respuesta puede ser enviada cuando el contenido solicitado ha sido borrado del servidor.

411 Length Required (en-US)

El servidor rechaza la petición porque el campo de encabezado Content-Length no esta definido y el servidor lo requiere.

412 Precondition Failed (en-US)

El cliente ha indicado pre-condiciones en sus encabezados la cual el servidor no cumple.

413 Payload Too Large

La entidad de petición es más larga que los límites definidos por el servidor; el servidor puede cerrar la conexión o retornar un campo de encabezado Retry-After.

414 URI Too Long (en-US)

La URI solicitada por el cliente es más larga de lo que el servidor está dispuesto a interpretar.

415 Unsupported Media Type (en-US) El formato multimedia de los datos solicitados no está soportado por el servidor, por lo cual el servidor rechaza la solicitud.

416 Requested Range Not Satisfiable (en-US)

El rango especificado por el campo de encabezado Range en la solicitud no cumple; es posible que el rango está fuera del tamaño de los datos objetivo del URI.

417 Expectation Failed (en-US)

Significa que la expectativa indicada por el campo de encabezado Expect solicitada no puede ser cumplida por el servidor.

418 I'm a teapot

El servidor se rehúsa a intentar hacer café con una tetera.

421 Misdirected Request

La petición fue dirigida a un servidor que no es capaz de producir una respuesta. Esto puede ser enviado por un servidor que no está configurado para producir respuestas por la combinación del esquema y la autoridad que están incluidos en la URI solicitada

422 Unprocessable Entity (en-US) (WebDAV (en-US))

La petición estaba bien formada pero no se pudo seguir debido a errores de semántica.

423 Locked (WebDAV (en-US))

El recurso que está siendo accedido está bloqueado.

424 Failed Dependency (WebDAV (en-US)) La petición falló debido a una falla de una petición previa.

426 Upgrade Required (en-US) El servidor se rehúsa a aplicar la solicitud usando el protocolo actual pero puede estar dispuesto a hacerlo después que el cliente se actualice a un protocolo diferente. El servidor envía un encabezado Upgrade en una respuesta para indicar los protocolos requeridos.

428 Precondition Required (en-US)

El servidor origen requiere que la solicitud sea condicional. Tiene la intención de prevenir problemas de 'actualización perdida', donde un cliente OBTIENE un estado del recurso, lo modifica, y lo PONE devuelta al servidor, cuando mientras un tercero ha modificado el estado del servidor, llevando a un conflicto.

429 Too Many Requests (en-US) El usuario ha enviado demasiadas solicitudes en un periodo de tiempo dado.

431 Request Header Fields Too Large (en-US)

El servidor no está dispuesto a procesar la solicitud porque los campos de encabezado son demasiado largos. La solicitud PUEDE volver a subirse después de reducir el tamaño de los campos de encabezado solicitados.

451 Unavailable For Legal Reasons (en-US) El usuario solicita un recurso ilegal, como alguna página web censurada por algún gobierno.

Errores de los servidores (500–599)

500 Internal Server Error

El servidor ha encontrado una situación que no sabe cómo manejarla.

501 Not Implemented (en-US)

...