

API Rest

Verbos HTTP

Son aquellos verbos propios del protocolo HTTP que fueron tomados para definir operaciones muy puntuales y específicas sobre los recursos de la API.

Los más utilizados son:

- **GET:** listado de recursos. Detalle de un solo recurso.
- **POST:** creación de un recurso.
- **PUT:** modificación total de un recurso.
- **PATCH:** modificación parcial de un recurso.
- **DELETE:** eliminación de un recurso. En muchas ocasiones es un soft delete, es decir, no se elimina definitivamente un recurso, sino que únicamente es marcado como eliminado o desactivado.

GET

GET es el tipo más simple de método de solicitud HTTP; La que usan los navegadores cada vez que hace clic en un enlace o escribe una URL en la barra de direcciones. Indica al servidor que transmita los datos identificados por la URL al cliente. Los datos nunca deben modificarse en el lado del servidor por una solicitud GET. En este sentido, una petición GET es de sólo lectura, pero por supuesto, una vez que el cliente recibe los datos, es libre de hacer cualquier operación con ella por su cuenta, por ejemplo, formatearla para su visualización.

PUT

Una petición PUT se utiliza cuando se desea crear o actualizar el recurso identificado por la URL. Por ejemplo,

PUT `http://api.com/clients/robin`

Podría crear un cliente, llamado Robin en el servidor. Usted notará que REST es completamente agnóstico de servidor; No hay nada en la solicitud que informe al servidor cómo deben crearse los datos, sólo que debería. Esto le permite intercambiar fácilmente la tecnología del servidor si la necesidad surge. Las peticiones PUT contienen los datos que se utilizarán para actualizar o crear el recurso en el cuerpo.

DELETE

DELETE debe realizar lo contrario de PUT; Debe utilizarse cuando desee eliminar el recurso identificado por la URL de la solicitud.

POST

POST se utiliza cuando el procesamiento que desea que suceda en el servidor debe repetirse, si la solicitud POST se repite (es decir, no son idempotent, más de esto a continuación). Además, las solicitudes POST deben causar el procesamiento del cuerpo de la solicitud como un subordinado de la URL que está publicando.

En palabras claras:

POST /clients/

No debe causar que el recurso en /clients/, se modifique, sino un recurso cuya URL comience con /clients/. Por ejemplo, podría agregar un nuevo cliente a la lista, con un ID generado por el servidor.

/clients/some-unique-id

Códigos de estado de respuesta HTTP

Los códigos de estado de respuesta HTTP indican si se ha completado satisfactoriamente una solicitud HTTP específica. Las respuestas se agrupan en cinco clases:

- Respuestas informativas (100)
- Respuestas satisfactorias (200)
- Redirecciones (300)
- Errores de los clientes (400)
- Errores de los servidores (500)

Respuestas informativas

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

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

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

Respuestas satisfactorias

- **GET:** El recurso se ha obtenido y se transmite en el cuerpo del mensaje.
- **HEAD:** Los encabezados de entidad están en el cuerpo del mensaje.
- **PUT o POST:** El recurso que describe el resultado de la acción se transmite en el cuerpo del mensaje.
- **TRACE:** El cuerpo del mensaje contiene el mensaje de solicitud recibido por el servidor.

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

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

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

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

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

226 IM Used

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 Multiple Choice

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

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

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 Obsoleto

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

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

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 cliente

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

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

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

407 Proxy Authentication Required

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

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

410 Gone

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

411 Length Required

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

412 Precondition Failed

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

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

415 Unsupported Media Type

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

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

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

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

423 Locked

El recurso que está siendo accedido está bloqueado.

424 Failed Dependency

La petición falló debido a una falla de una petición previa.

426 Upgrade Required

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

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

El usuario ha enviado demasiadas solicitudes en un periodo de tiempo dado.

431 Request Header Fields Too Large

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

El usuario solicita un recurso ilegal, como alguna página web censurada por algún gobierno.

Errores de servidor

500 Internal Server Error

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

501 Not Implemented

El método solicitado no está soportado por el servidor y no puede ser manejado. Los únicos métodos que los servidores requieren soporte (y por lo tanto no deben retornar este código) son GET y HEAD.

502 Bad Gateway

Esta respuesta de error significa que el servidor, mientras trabaja como una puerta de enlace para obtener una respuesta necesaria para manejar la petición, obtuvo una respuesta inválida.

503 Service Unavailable

El servidor no está listo para manejar la petición. Causas comunes puede ser que el servidor está caído por mantenimiento o está sobrecargado. Hay que tomar en cuenta que junto con esta respuesta, una página usuario-amigable explicando el problema debe ser enviada. Estas respuestas deben ser usadas para condiciones temporales y el encabezado HTTP Retry-After: debería, si es posible, contener el tiempo estimado antes de la recuperación del servicio. El webmaster debe también cuidar los encabezados relacionados al caché que son enviados junto a esta respuesta, ya que estas respuestas de condición temporal deben usualmente no estar en el caché.

504 Gateway Timeout

Esta respuesta de error es dada cuando el servidor está actuando como una puerta de enlace y no puede obtener una respuesta a tiempo.

505 HTTP Version Not Supported

La versión de HTTP usada en la petición no está soportada por el servidor.

506 Variant Also Negotiates

El servidor tiene un error de configuración interna: negociación de contenido transparente para la petición resulta en una referencia circular.

507 Insufficient Storage

El servidor tiene un error de configuración interna: la variable de recurso escogida está configurada para acoplar la negociación de contenido transparente misma, y no es por lo tanto un punto final adecuado para el proceso de negociación.

508 Loop Detected

El servidor detectó un ciclo infinito mientras procesaba la solicitud.

510 Not Extended

Extensiones adicionales para la solicitud son requeridas para que el servidor las cumpla.

511 Network Authentication Required

El código de estado 511 indica que el cliente necesita autenticar para obtener acceso a la red.

Referencias

Coppola, M. (2023, 21 enero). Qué es una API REST, para qué sirve y ejemplos. blog.hubspot.es. Recuperado 5 de febrero de 2024, de <https://blog.hubspot.es/website/que-es-api-rest>

Fischer, L. (2022, 13 febrero). Guía para principiantes de HTTP y REST. Code Envato Tuts+. <https://code.tutsplus.com/es/a-beginners-guide-to-http-and-rest--net-16340t>

Códigos de Estado de respuesta HTTP - HTTP | MDN. (2022, 26 noviembre). MDN Web Docs. <https://developer.mozilla.org/es/docs/Web/HTTP/Status>