

Métodos o verbos HTTP REST

Métodos HTTP

HTTP define una serie predefinida de métodos de petición (algunas veces referido como "verbos") que pueden utilizarse. Cada método indica la acción que desea que se efectúe sobre el recurso identificado. Lo que este recurso representa depende de la aplicación del servidor.

GET

El método GET solicita una representación del recurso especificado. Las solicitudes que usan GET solo deben recuperar datos y no deben tener ningún otro efecto.

Sintaxis: GET /index.html

POST

El método POST envía datos al servidor. El tipo del cuerpo de la solicitud es indicada por la cabecera Content-Type. Una solicitud POST es típicamente enviada por un formulario HTML y resulta en un cambio en el servidor. En este caso, el tipo de contenido es seleccionado poniendo la cadena de texto adecuada en el atributo method del elemento o el atributo submit de los elementos o Cuando la solicitud POST es enviada por otro método distinto a un formulario HTML, por ejemplo mediante una XMLHttpRequest, el cuerpo puede aceptar cualquier tipo.

Sintaxis: POST /index.html

PUT

La petición PUT crea un nuevo elemento o reemplaza una representación del elemento de destino con los datos de la petición. La diferencia entre el método PUT y el método POST es que PUT es un método idempotente: llamarlo una o más veces de forma sucesiva tiene el mismo efecto (sin efectos secundarios), mientras que una sucesión de peticiones POST idénticas pueden tener efectos adicionales, como enviar una orden varias veces.

Sintaxis: PUT /nuevo.html HTTP/1.1

DELETE

El método DELETE solicita que el servidor de origen elimine la asociación entre el recurso de destino y su funcionalidad actual. Si se aplica con éxito un método DELETE, el servidor de origen DEBERÍA enviar:

- Un código de estado 202 (Aceptado) si la acción probablemente tendrá éxito pero aún no se ha promulgado,
- Un código de estado 204 (Sin contenido) si la acción se ha llevado a cabo y no se debe proporcionar más información, o
- Un código de estado 200 (OK) si la acción se ha realizado y el mensaje de respuesta incluye una representación que describe el estado.

Sintaxis: DELETE /file.html HTTP/1.1

HEAD

El método HEAD pide una respuesta idéntica a la que correspondería a una petición GET, pero en la respuesta no se devuelve el cuerpo. Esto es útil para poder recuperar los metadatos de los encabezados de respuesta, sin tener que transportar todo el contenido.

Sintaxis: HEAD /index.html 6

PATCH

Su función es la misma que PUT, el cual sobrescribe completamente un recurso. Se utiliza para actualizar, de manera parcial una o varias partes. El método HTTP PUT únicamente permite reemplazar completamente un documento. A diferencia de PUT, el método PATCH no es idempotente, esto quiere decir que peticiones idénticas sucesivas pueden tener efectos diferentes. Sin embargo, es posible emitir peticiones PATCH de tal forma que sean idempotentes.

PATCH (al igual que POST) puede provocar efectos secundarios a otros recursos.

Sintaxis: PATCH /file.txt HTTP/1.1

OPTIONS

Devuelve los métodos HTTP que el servidor soporta para un URL específico. Esto puede ser utilizado para comprobar la funcionalidad de un servidor web mediante petición en lugar de un recurso específico.

Solicita opciones de comunicación permitidas para una URL o servidor determinado. Un cliente puede especificar una URL con este método o un asterisco (*) para hacer referencia a todo el servidor.

Sintaxis: OPTIONS /index.html HTTP/1.1
Sintaxis: OPTIONS * HTTP/1.1

TRACE

Este método solicita al servidor que introduzca en la respuesta todos los datos que reciba en el mensaje de petición. Se utiliza con fines de depuración y diagnóstico ya que el cliente puede ver lo que llega al servidor y de esta forma ver todo lo que añaden al mensaje los servidores intermedios.

Realiza una prueba de bucle invertido de mensajes a lo largo de la ruta al recurso de destino, lo que proporciona un mecanismo de depuración útil. El destinatario final de la solicitud debe reflejar el mensaje recibido, excluyendo cualquier campo que pueda incluir datos confidenciales, al cliente como el cuerpo del mensaje de una respuesta 200.

Tenga en cuenta que el cliente no debe enviar contenido en la solicitud ni generar campos que puedan incluir datos confidenciales, como credenciales de usuario almacenadas o cookies.

Sintaxis: TRACE /index.html

CONNECT

Se utiliza para saber si se tiene acceso a un host, no necesariamente la petición llega al servidor, este método se utiliza principalmente para saber si un proxy nos da acceso a un host bajo condiciones especiales, como por ejemplo "corrientes" de datos bidireccionales encriptadas (como lo requiere SSL).

Sintaxis: www.example.com:443 HTTP/1.1

REFERENCIAS:

Métodos de Petición y Códigos de devolución

https://es.wikipedia.org/wiki/Protocolo_de_transferencia_de_hipertexto

GET Method

<https://developer.mozilla.org/es/docs/Web/HTTP/Methods/GET>

PUT Method

<https://developer.mozilla.org/es/docs/Web/HTTP/Methods/PUT>

<https://www.rfc-editor.org/rfc/rfc9110#PUT>

POST Method

<https://developer.mozilla.org/es/docs/Web/HTTP/Methods/POST>

DELETE Method

<https://developer.mozilla.org/es/docs/Web/HTTP/Methods/DELETE>

HEAD Method

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods/HEAD>

PATCH Method

<https://developer.mozilla.org/es/docs/Web/HTTP/Methods/PATCH>

OPTIONS Method

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods/OPTIONS>

CONNECT Method

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods/CONNECT>