



#YoProgramo

← Volver al menu principal

Material de lectura

Introducción a Desarrollo Web y Aplicaciones

HyperText Transfer Protocol (HTTP)

Hypertext Transfer Protocol ([HTTP](#)) (Protocolo de Transferencia de Hipertexto en español) es un protocolo de la capa de aplicación para la transmisión de documentos hipermedia, como HTML. Fue diseñado para la comunicación entre los navegadores y servidores web, aunque puede ser utilizado para otros propósitos también.

HTTP define un conjunto de [métodos de petición](#) para indicar la acción que desea realizar para un recurso determinado. Aunque estos también pueden ser sustantivos, estos métodos de solicitud a veces son llamados *verbos HTTP*.

Ayuda

Texto a voz



Es muy importante saber que HTTP es un protocolo sin estado, es decir, no guarda ninguna información sobre conexiones anteriores. Luego veremos las implicancias de esto.

Una descripción importante del protocolo es que **está orientado a transacciones y sigue el esquema de petición / respuesta entre un cliente y un servidor**. El cliente realiza una petición enviando un mensaje con cierto formato al servidor. El servidor le envía un mensaje de respuesta. Para hacerlo más concreto, un cliente podría ser un navegador web y un servidor podría ser una aplicación en un servidor web corriendo en Internet.

Para que entendamos la forma de cómo se programa una aplicación web, necesitamos en alguna medida entender los mensajes HTTP. Los mensajes HTTP son en texto plano lo que hace más legible y fácil de depurar. Estos tienen la siguiente estructura:

.- Primero, hay una línea inicial en donde se diferencian dependiendo de si son peticiones y respuestas. Para las solicitudes la línea comienza con una acción requerida por el servidor, a esto se le denomina método de petición seguido de la url del recurso y la versión http que soporte al cliente. Lo importante es el método de petición y la URL (Uniform Resource Locator o localizador de recursos uniforme).

.- Para las respuestas, la línea comienza con la versión de HTTP seguido por un código de respuesta y con una frase asociada a dicho retorno.

.- También los mensajes tienen una cabecera que son metadatos con información diversa y el cuerpo de mensaje que es opcional. Típicamente este cuerpo tiene los datos que se intercambian entre el cliente y el servidor.

Los métodos de petición (o también llamados verbos) son varios. Cada método indica la acción que desea que se efectúe sobre el recurso identificado lo que este recurso representa dependiente de la aplicación del servidor.

Los métodos que vamos a usar son los de **get** y **post**. Cabe destacar que hay convenciones en donde se utilizan otros métodos. En este punto entran en juego el estándar REST y las API's.

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

[Ayuda](#)[Texto a voz](#)



El método post envía los datos para que sean procesados por el recurso identificado. Los datos enviados se incluirán en el de la petición. Esto puede resultar en la creación de un nuevo recurso o de las actualizaciones de los recursos existentes.

Para finalizar con la explicación de HTTP vamos a nombrar códigos de respuesta:

1. Por ejemplo, el 200 representa una respuesta correcta es decir que indica que la petición ha sido procesada correctamente.
2. Otro ejemplo es la respuesta 404 que significa errores causados por el cliente, por ejemplo que el recurso no se encuentra.
3. Finalmente, las respuestas que comienzan con 5 por ejemplo el 500 son errores causados por el servidor. Esto indica que ha habido un error en el proceso de la petición a causa de un fallo en el servidor.

Todo lo que hablamos de HTTP lo vamos a usar al momento de programar una aplicación web. Vamos a ver que al presionar por ejemplo un enlace se hace una petición “get” al servidor para buscar otra página lo que resultará en una respuesta 200 si se encuentra la página o en un 404 si no se encuentra. También veremos que al llenar un formulario y enviarlo al servidor lo haremos a través de una petición post.

Los otros protocolos de TCP / IP son necesarios para configuraciones de conexiones, pero a nivel de aplicación no los utilizaremos. Eso no significa que no haya que entenderlos pues nos facilitarán la solución de problemas, por ejemplo, al conectarnos a una base de datos desde la aplicación web del servidor.

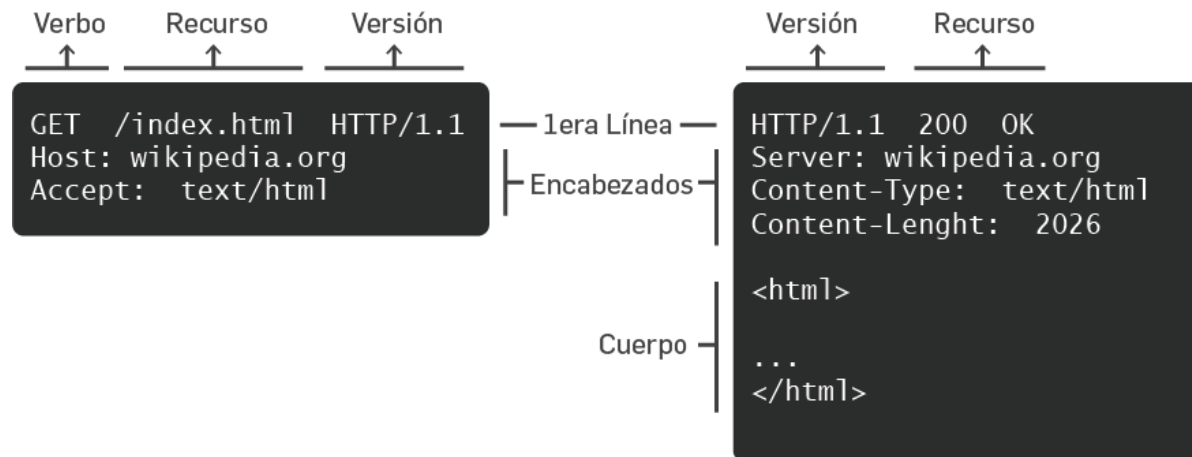


Ayuda

Texto a voz

PETICIÓN

RESPUESTA



Métodos de petición HTTP - HTTP | MDN. (s. f.). Mozilla.Org. Recuperado 11 de octubre de 2021, de <https://developer.mozilla.org/es/docs/Web/HTTP/Methods>

Method	Description	Sec.
GET	Transfer a current representation of the target resource.	4.3.1
HEAD	Same as GET, but only transfer the status line and header section.	4.3.2
POST	Perform resource-specific processing on the request payload.	4.3.3
PUT	Replace all current representations of the target resource with the request payload.	4.3.4
DELETE	Remove all current representations of the target resource.	4.3.5
CONNECT	Establish a tunnel to the server identified by the target resource.	4.3.6
OPTIONS	Describe the communication options for the target resource.	4.3.7
TRACE	Perform a message loop-back test along the path to the target resource.	4.3.8

rfc7231. (s. f.). Ietf.Org. Recuperado 11 de octubre de 2021, de <https://datatracker.ietf.org/doc/html/rfc7231#section-4>

Ayuda

Texto a voz

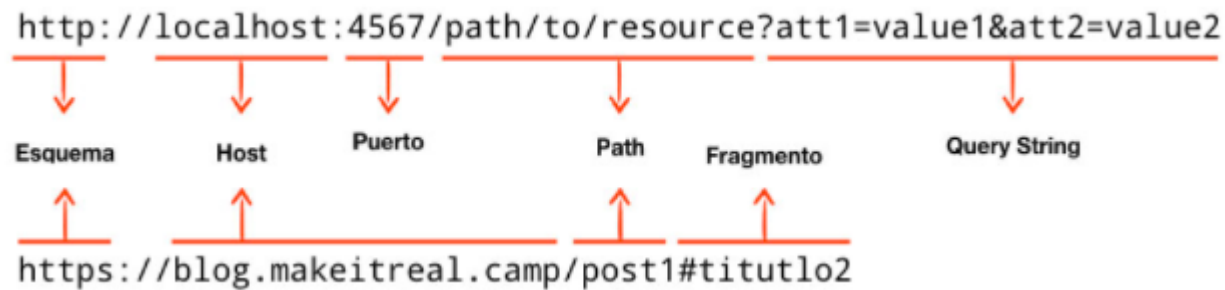
Verbos y cómo se relacionan con operaciones

Código de estados

Cuando el servidor devuelve una respuesta se indica un código de estado:

- **1xx:** Mensaje informativo.
- **2xx:** Exito
 - 200 OK
 - 201 Created
 - 202 Accepted
 - 204 No Content
- **3xx:** Redirección
 - 300 Multiple Choice
 - 301 Moved Permanently
 - 302 Found
 - 304 Not Modified
- **4xx:** Error del cliente
 - 400 Bad Request
 - 401 Unauthorized
 - 403 Forbidden
 - 404 Not Found
- **5xx:** Error del servidor
 - 500 Internal Server Error
 - 501 Not Implemented
 - 502 Bad Gateway
 - 503 Service Unavailable

Código de respuesta HTTP:



URL: Composición de la URL

Ayuda

Texto a voz



Actividad previa

◀ Transferencia de datos

Siguiente actividad

Glosario ▶

📁 Resumen de retención de datos



Ayuda

Texto a voz