

Generalidades del protocolo HTTP

ERIK JARQUÍN SÁNCHEZ

erikjarquin20@gmail.com

14 de Marzo del 2022

1. Introducción

Este paquete se va a centrar en hablar sobre las generalidades del protocolo HTTP, este protocolo; que es un protocolo de transferencia de hipertexto, usado en la tecnología de redes, páginas web, entre otras muchas. Por lo tanto es muy importante conocer sus funcionalidades básicas, así como los tipos de requisitos que solicita para su funcionamiento o los tipos de componentes que posee y como funcionan estos mismos.

2. Desarrollo

2.1. Protocolo HTTP

El Protocolo de Transferencia de Hipertexto (Hypertext Transfer Protocol) es un sencillo protocolo cliente-servidor que articula los intercambios de información entre los clientes Web y los servidores HTTP. La especificación completa del protocolo HTTP 1/0 está recogida en el RFC 1945. Fue propuesto por Tim Berners-Lee, atendiendo a las necesidades de un sistema global de distribución de información como el World Wide Web.[2]

2.2. Mensajes

El protocolo HTTP cuenta con dos tipos de mensajes de “solicitud y de respuesta”, donde los mensajes de solicitud van del cliente hasta el servidor y los de respuesta del servidor al cliente.

Los mensajes de solicitud y de respuesta hacen uso de un formato especial para la transferencia de los mismos, dicho formato consta de una línea con cero o más cabeceras (donde cada una de ellas contiene sus propios campos), seguido de una línea vacía (CRLF) para posteriormente comenzar con el cuerpo del mensaje.[4]

2.3. Métodos de petición

En la tabla 1 se verán los métodos de petición que utiliza el protocolo HTTP, según la fuente donde se busco dicha información se lista 8 métodos. HTTP define 8 métodos (algunas veces referido como "verbos") que indica la acción que desea que se efectúe sobre el recurso identificado. Lo que este recurso representa, si los datos pre-existentes o datos que se generan de forma dinámica, depende de la aplicación del servidor. A menudo, el recurso corresponde a un archivo o la salida de un ejecutable que residen en el servidor.[3]

Método	Significado
GET	Devuelve el recurso identificado en la URL pedida.
HEAD	Funciona como el GET, pero sin que el servidor devuelva el cuerpo del mensaje. Es decir, sólo se devuelve la información de cabecera.
POST	Indica al servidor que se prepare para recibir información del cliente. Suele usarse para enviar información desde formularios.
PUT	Envía el recurso identificado en la URL desde el cliente hacia el servidor.
OPTIONS	Pide información sobre las características de comunicación proporcionadas por el servidor. Le permite al cliente negociar los parámetros de comunicación.
TRACE	Inicia un ciclo de mensajes de petición. Se usa para depuración y permite al cliente ver lo que el servidor recibe en el otro lado.

Método	Significado
DELETE	Solicita al servidor que borre el recurso identificado con el URL.
CONNECT	Este método se reserva para uso con proxys. Permitirá que un proxy pueda dinámicamente convertirse en un túnel. Por ejemplo para comunicaciones con SSL.

Tabla 1: Tabla de métodos de petición

2.4. Cabeceras

En la tabla 2 se muestran algunas de las cabeceras más comunes o más generales del protocolo HTTP, pero no son las únicas cabeceras, hay más, especificadas para diferentes acciones que realizan funciones específicas.

Se llaman así porque tienen una aplicación general para los mensajes de solicitud y de respuesta, pero que no aplica a la entidad que es transferida. Estos campos sólo se aplican al mensaje que se transmite [4]

Nombre de la Cabecera	Descripción de la Cabecera
Cache-Control	Se utiliza para especificar directivas, las cuales deben ser obedecidas por todos los mecanismos de almacenamiento en caché, a lo largo de las peticiones de solicitud/respuesta para almacenamiento en caché, por medio de algoritmos predeterminados por las mismas directivas
Connection	Esta cabecera le permite al remitente realizar especificaciones para conexiones especiales, sin necesidad de comunicarse a los proxies con conexiones adicionales.
Date	Se encarga de especificar la fecha y la hora en que se originó el mensaje. Un ejemplo es: Date: Tue, 15 Nov 1994 08:12:31 GMT

Nombre de la Cabecera	Descripción de la Cabecera
Pragma	Es usado para incluir directivas específicas de implementación que podrían aplicarse a lo largo de la cadena de solicitud/respuesta. Estas generalmente, especifican el punto de vista del protocolo.
Trailer	Éste indica el conjunto de campos de cabeceras, que está presente en el trailer de un mensaje codificado, con una codificación de transferencia fragmentada.
Transfer-Encoding	Indican los tipos de transformaciones que ha sufrido el cuerpo del mensaje, con el fin de llevar a cabo una transferencia segura entre el remitente y el destinatario.
Upgrade	Se encarga de que el cliente especifique qué protocolos de comunicación está utilizando, para apoyarse en la comunicación, así como que protocolo desea utilizar si el servidor considera adecuado cambiar de protocolo.
Via	Generalmente es utilizado por los Gateway y los proxies, para indicar los protocolos intermedios y los beneficiarios entre el usuario agente y el servidor de peticiones, y entre el servidor de origen y las respuestas del cliente.
Warning	Es un campo de alerta que se utiliza para transportar información, acerca de la situación o de la transformación del mensaje, advirtiendo de posibles fallas de las operaciones o transformaciones que se le han aplicado el cuerpo del mensaje.

Tabla 2: Tabla de cabeceras

2.5. Ejemplos de dialogo HTTP

Las peticiones POST suelen emplearse con formularios digitales.[5] Abajo encontrarás el ejemplo de un formulario que recoge un nombre y una dirección de correo electrónico y lo envía al servidor

por medio de POST:

```
<html>
<body>
<form action="newsletter.php" method="post">
Name: <input type="text" name="name"><br>
E-mail: <input type="text" name="email"><br>
<input type="submit">
</form>
</body>
</html>
```

La respuesta del servidor está formada por encabezados seguidos del recurso solicitado, en el caso de una página web: [1]

```
HTTP/1.1 200 OK
Date: Fri, 31 Dec 2003 23:59:59 GMT
Content-Type: text/html
Content-Length: 1221
<html lang="eo">
<head>
<meta charset="utf-8">
<title>Título del sitio</title>
</head>
<body>
<h1>Página principal de tuHost</h1>
(Contenido)
.
.
.
</body>
</html>
```

3. Conclusiones

En este apartado hablamos sobre el protocolo HTTP, este tiene muchos beneficios para poder navegar en la red, pero también cuenta con una que otra desventaja. En si tiene muchos tipos de especificaciones y características que ayudan a que pueda tener un buen funcionamiento y funcionar en la red. Nos damos cuenta que el protocolo es muy importante en mantener nuestra navegación en internet funcional de muchas maneras.

Referencias

- [1] Developer.mozilla.org. 2022. códigos de estado de respuesta http - http — mdn. [online] available at: <https://developer.mozilla.org/es/docs/web/http/status> [accessed 14 march 2022].
- [2] Generalidades del protocolo http - http — mdn. (2022). retrieved 15 march 2022, from <https://developer.mozilla.org/es/docs/web/http/overview>.
- [3] Protocolo http. retrieved 15 march 2022, from https://alvaroprimoguijarro.files.wordpress.com/2012/01/ud04_http_alvaroprimoguijarro.pdf.
- [4] Protocolos y software en el servicio web. <https://hopelchen.tecnm.mx/principal/sylabus/fpdb/recursos/r130647.pdf>.
- [5] técnicas, c., request, h. (2022). http request: los métodos de petición que debes conocer. retrieved 15 march 2022, from <https://www.ionos.mx/digitalguide/hosting/cuestiones-tecnicas/http-request/>.