Generalidades del protocolo HTTP

Leonel Alexis García Juárez
alexrez2000@gmail.com
Universidad de la Sierra Sur

2022/03/14

1. Generalidades del protocolo HTTP

El protocolo de transferencia de Hipertexto (HTTP) es un protocolo cliente-servidor, este se encarga de los intercambios de información entre los clientes web y los servidores HTTP. Este protocolo fue propuesto por Tim Berners-Lee (1999).
HTTP es soportado por servidores de conexión TCP/IP además de funcionar de igual manera en entornos UNIX. HTTP hace uso del protocolo TCP (Protocolo de Control de Transmisión) para establecer y mantener una comunicación asegurando el intercambio de los datos de manera segura, HTTP es un protocolo de aplicación.
El protocolo de HTTP se basa en cliente-servidor de esta manera el agente usuario (proxy) realiza la petición, estas peticiones son recibidas por un servidor el cual gestiona y responde.

2. Mensajes

El protocolo HTTP/1.1 y sus versiones anteriores eran de formato de texto y totalmente comprensibles, en HTTP/2 los mensajes están estructurados enn un nuevo formato binario, de esta manera las tramas permiten la comprensión de las cabeceras y su multiplexación, esto provoca que si solamente un parte del mensaje es enviado será en este formato, haciendo que la semántica de cada mensaje sea la misma. Existen dos tipos de mansajes en el protocolo HTTP: peticiones y respuestas cada uno cuenta con su propio formato.

Peticiones
Los campos que componen un mensaje de petición son:

Un método HTTP: Este campo define la operación que el cliente quiere realizar, el cliente suele pedir recursos, o presentar un valor en un formulario HTL haciendo uso por ejemplo GET, POS, OPTIONS, HEAD, etc.
La dirección del recurso pedido: El URL del recurso
La versión del protocolo HTTP
Cabeceras HTTP opcionales: Aportan mayor información a los servidores

Cuerpo del mensaje

Respuestas Se forman por los campos: Version del protocolo HTTP Código de estado (exitoso, fallido, etc.) Mensaje de estado Cabeceras HTTP El recurso que se ha pedido.

3. Métodos de petición

HTTP define 8 métodos que indican la acción que se quiere realizar, y el servidor responde a un archivo o las salidas de un ejecutable que se encuentre en el servidor. Los métodos que define HTTP son:

GET: sirve para obtener la URL correspondiente en una línea de petición

HEAD: pide una respuesta idéntica a las+ que correspondería a una petición GET, pero lo devuelto no se envía al cuerpo si no al encabezado.

POST: sirve para enviar una entidad que el servidor tiene que incorporar en el recurso identificado por el URL de la línea de petición.la semántica depende de :

• Añadir contenido a un recurso existente,

• Mandar un mensaje a un grupo de noticias

• Crear un registro nuevo en una base de datos

• Pasar datos a un programa que se tiene que ejecutar al servidor. Un caso típico de este último ejemplo son los datos de un formulario HTML.

PUT y DELETE: son ocupados para actualizar y borrar:

4. Códigos de respuesta

Los códigos se establecen para mandar un mensaje por ejemplo: respuestas informativas, respuestas satisfactorias, redirecciones, errores de los clientes y errores de los servidores. Los principales códigos de respuesta son:

1xx son de tipo informativas.

2xx son satisfactorias.

3xx son redirecciones.

4xx error en cliente, (404 *Not Found*)

5xx error en el servidor, (500 *Internal Server Error* significa que el servidor ha sido incapaz de procesar la petición tal vez por un error de programación o falta de recursos).

5. Cabeceras

Principales cabeceras en peticiones:

Host: Se especifica el nombre del dominio del servidor y opcionalmente el puerto TCP en el que el servido está escuchando, si no se elige un puerto se ocupa el predeterminado. **User-Agent:** contiene una cadena que permite a los pares del protocolo de red identificar el tipo de aplicaciòn, el S.O., el provedor del software o la versión de este.

Cabeceras en respuestas

Cache-control: Se usa para especificar directivas que deben ser obedecidas por los mecas¡nismos de cacheo junto a la petición.

Content-type: Esta cabecera indica el "mime-type" del documento. Se decide como se interpreta los contenidos.

6. Ejemplo

ejemplo 1:

<dialog open>

¡Saludos a todos y todas!

</dialog>

ejemplo2:

<button type="button.ºnclick=.ªbrirDialogo('dialogo1')">Abrir diálogo</button>

<dialog id="dialogo1">

Hola! Este es nuestro primer diálogo. Si te gusta, no lo cierres...

<button type="button.ºnclick=çerrarDialogo('dialogo1')">Cerrar</button>

</dialog>

7. Bibliografía

Referencias

[1] *Generalidades del protocolo HTTP - HTTP — MDN.* (2020, 8 diciembre). MDn. Recuperado 10 de marzo de 2022, de https://developer.mozilla.org/es/docs/Web/HTTP/Overview

[2] *Introducción a HTTP/2 — Web Fundamentals —.* (s. f.). Google Developers. Recuperado 10 de marzo de 2022, de https://developers.google.com/web/fundamentals/performance/http2/?hl=es

[3] *Protocolo HTTP.* (2020, 5 noviembre). victorponz. Recuperado 10 de marzo de 2022, de https://victorponz.github.io/Ciberseguridad-PePS/tema1/http/2020/11/05/Protocolo-HTTP.html