

HTTP: Hypertext Transfer Protocol

- Base de cualquier intercambio de datos en la Web
- Protocolo estructura cliente-servidor
- Por ejemplo una pagina web completa resulta la union de diferentes documentos recibidos, por ejemplo un css, un js, imagenes y el html
- Funciona en capa de aplicacion, se transmite sobre protocolo TCP o TLS (encriptado) de capa 4.

Se utiliza tanto para las webs como para imagenes, videos, datos de formularios, etc.

Arquitectura

Por lo general, un navegador web (agente) realiza una peticion a un servidor web. Este ultimo gestiona y responde.

1. El navegador solicita un documento HTML al servidor
2. Procesa el documento recibido y envia mas peticiones para scripts, estilos, videos, imagenes, etc.
3. Une todos los datos y compone la pagina final.

En cuanto al servidor, pueden ser multiples servidores en un unico computador o un servidor puede estar repartido en multiples ordenadores. (estandar http/1.1)

Proxies

En la estructura de la web, existen dispositivos entre el cliente y el servidor que gestionan los mensajes.

- La mayoría de estos dispositivos gestionan niveles de protocolo inferiores tales como transporte, red o fisica.
- Los dispositivos que operan en capa de aplicacion (mayor procesamiento) por ende gestionan HTTP, son conocidos como proxies. Entre sus funciones destacamos
 - caching (publica o privada)
 - filtrado (anti-virus, control parental)
 - balanceo de carga
 - autentificacion
 - registro de eventos

Características de HTTP

- Pensado para ser leído e interpretable por personas
- Extensible
- Utiliza sesiones, pero no estados.
 - No guarda ningun dato entre dos peticiones en la misma sesion.
 - Se permite guardar datos con respecto a la sesion de comunicacion, mediante cookies
 - Requiere trabajar sobre un protocolo fiable de comunicacion, por eso utiliza TCP en lugar de UDP.

Flujo HTTP

1. Cliente abre conexion TCP mediante una o varias peticiones al servidor

2. Se realiza una petición HTTP
3. Se lee la respuesta del servidor, la cual consiste en información necesaria para la conexión seguida del contenido, por ejemplo el archivo html.

Mensajes HTTP

En HTTP/1.1 los mensajes eran en formato texto y legibles por las personas. Sin embargo, en HTTP/2 los mensajes son en formato binario. Sin embargo la semántica se mantiene y se puede interpretar los mensajes de HTTP/2 en formato HTTP/1.1

Petición HTTP

- Método: Define la operación que el cliente quiere realizar
- Dirección del recurso
- Versión del protocolo HTTP
- Cabeceras HTTP opcionales
- Cuerpo de mensaje (de ser necesario en el método pedido)

Respuestas

- Versión del protocolo HTTP
- Código de estado
- Mensaje de estado
- Cabeceras HTTP
- Opcionalmente el recurso pedido

Recursos Web

El objetivo de una petición HTTP es llamada recurso. Puede ser una foto, un documento, etc. Cada recurso se identifica mediante un URI (Uniform Resource Identifier)

URLs

- Forma más común de URI es la Uniform Resource Locator (URL) también conocida como dirección web.
- gianfrancolasala.com/blog

URNs

Uniform Resource Name (URN) identifica un recurso por un nombre en particular

Sintaxis

- <http://www.gianfrancolasala.com:80/path/to/some/file.html?key1=value1&key2=value2#somewhere>
- http:// -> Protocolo que el buscador debe utilizar
 - data, file, ftp, http/https, javascript, mailto, ssh, tel, urn, etc.
- www.example.com: Nombre de dominio o autoridad a la cual pertenece el nombre. Es posible acceder directamente la dirección IP.
- :80: puerto. Si el servidor web utiliza el puerto estándar (80 para HTTP y 443 para HTTPS) no es requerido.

- /path/to/some/image.html -> path/direccion del recurso dentro del servidor web.
- Query -> Parametros entregados al servidor de pares llave/valor.
- #Somewhere -> Fragmento. Representa una marca dentro del recurso En un documento es posible que el navegador scrollee hasta la parte mencionada
-

Tipos MIME

- Media Type (Multipurpose Internet Mail Extensions or MIME Type)
- Forma de indicar el formato de un documento, archivo o conjunto de datos [RFC6838]
- Los navegadores utilizan el MIME type para determinar como se procesa un documento
- NO utilizan la extension del archivo

Sintaxis

```
tipo/subtipo  
  
text/plain  
text/html  
image/jpeg  
image/gif  
video/mp4  
audio/*
```

Tipos Discretos

- Indican categoria del documento
- text, image, audio, application (datos binarios), video

Tipos multiparte

indican documentos que se encuentran en partes, posiblemente con diferentes tipos de MIME. De esta forma se representan documentos compuestos.

multipart/form-data

El tipo ``multipart/form-data`` se utiliza para enviar contenido de un formulario HTML completo desde el browser al sv.

MIME Sniffing

La ausencia de tipo MIME hace que algunos navegadores adivinen el tipo correcto observando el recurso. Esta practica afecta la seguridad ya que algunos tipos MIME representan contenido ejecutable