



APRENDERAPROGRAMAR.COM

HEADERS CABECERAS
HTTP. QUÉ SON Y PARA
QUÉ SIRVEN. REQUEST Y
STATUS LINE. REFERER,
AUTHORIZATION, USER-
AGENT... (CU01208F)

Sección: Cursos

Categoría: Tutorial básico del programador web: Ajax desde cero

Fecha revisión: 2031

Resumen: Entrega nº8 del Tutorial básico "Ajax desde cero".

Autor: Alex Rodríguez

¿QUÉ SON LAS CABECERAS O HEADERS HTTP?

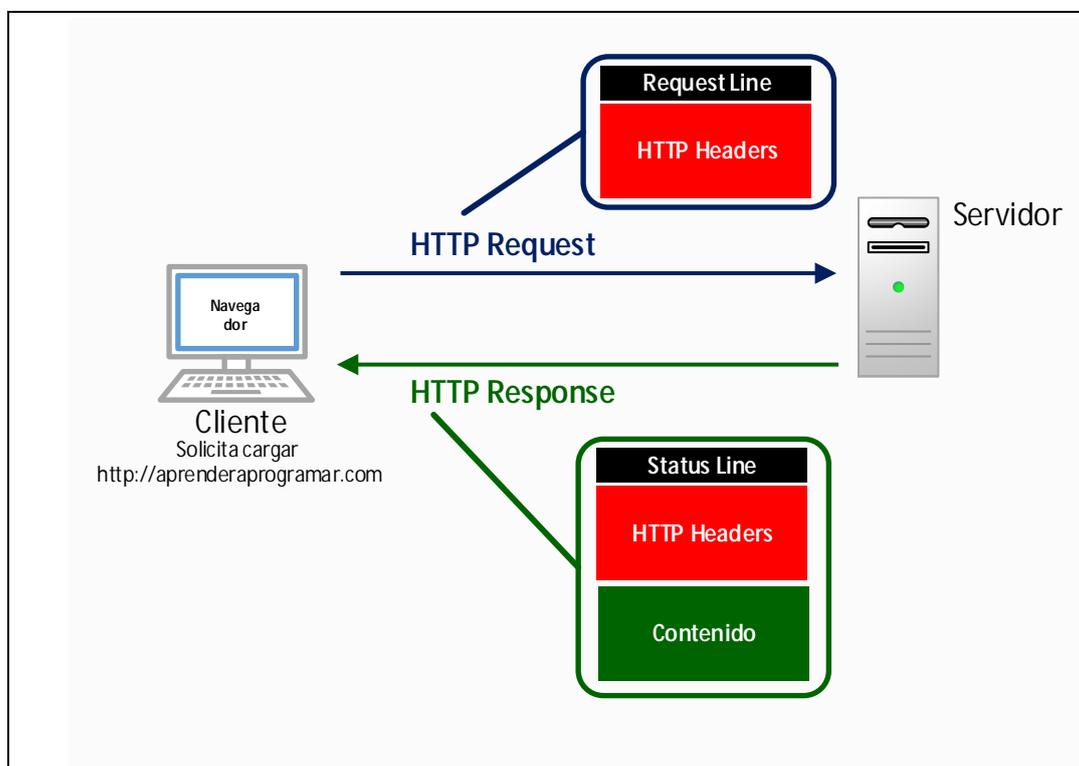
Antes de estudiar los métodos del objeto XMLHttpRequest vamos a comentar brevemente algo que nos va a ser útil para mejorar nuestra comprensión de ellos. ¿Qué son las cabeceras HTTP? En general una dirección web comienza con http://... Donde http significa "HyperText Transfer Protocol". Este es un protocolo que usa todo internet para las comunicaciones desde los años 90.



Con este protocolo, la carga de una página web se basa en peticiones HTTP que envía el navegador al servidor y en respuestas HTTP que envía el servidor al navegador. Al cargar una página web se producen decenas de peticiones y respuestas HTTP cada una de las cuales lleva cabeceras (headers). Las cabeceras llevan información necesaria para la comunicación y pueden incluir diferentes aspectos como: tipo de navegador que realiza la petición, dirección de la página solicitada, juego de caracteres utilizado, etc.

PETICIONES Y RESPUESTAS HTTP

La carga de una página web comprende decenas de peticiones al servidor y respuestas de este cuyo esquema básico es el siguiente:



CABECERAS DE PETICIONES (REQUEST)

Una petición al servidor consta de una línea de petición que contiene alguna información básica sobre la petición y de diversas líneas que constituyen los headers. Un ejemplo puede ser:

```
GET /index.php?option=com_content&view=article&id=57&Itemid=86 HTTP/1.1
Host: aprenderaprogramar.com
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; rv:42.0) Gecko/20100101 Firefox/42.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: es-ES,es;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://aprenderaprogramar.com/
Cookie: 6bat36d0999f4da0e8214bc3a90328f50=p3eug5t?pvonb5lrf40jf5dcf71
Connection: keep-alive
```

La primera línea es la request line y el resto son cabeceras HTTP.

La request line está formada por 3 partes: el método, la ruta y el protocolo.

El método indica el tipo de petición que se realiza, y que típicamente puede ser GET, POST ó HEAD. La carga de una web normalmente se hace por GET. Sin embargo, cuando por ejemplo enviamos un formulario es habitual que la petición se haga por post, de modo que la línea de petición podría ser de la forma POST /resultados.php HTTP 1.1. También el envío de un formulario puede hacerse por GET, en este caso bajo un formato similar a GET /form.php?nombre=Juan&apellidos=Perez&action=Submit HTTP/1.1

El método HEAD es similar a GET pero con la diferencia de que el servidor no envía el contenido con la respuesta, sino simplemente la línea de status y las cabeceras. Esto sirve por ejemplo para comprobar si un link está operativo (responde) sin necesidad de recuperar todo el contenido, lo cual hace que se puedan hacer chequeos mucho más rápidos que usando get.

La ruta, es normalmente la ruta relativa o parte de la dirección web que viene detrás del dominio. Finalmente el protocolo consta de HTTP y el número de versión de protocolo que se emplea, típicamente 1.1.

El resto de cabeceras http son pares "Nombre: valor", por ejemplo, Host: aprenderaprogramar.com. Estas cabeceras contienen información diversa sobre la petición HTTP y sobre el navegador. Por ejemplo la línea de User-Agent proporciona información sobre el navegador y sistema operativo desde el que se hace la petición, y Accept-Encoding informa al servidor si el navegador puede aceptar datos comprimidos bajo formatos como gzip.

La mayoría de las cabeceras HTTP son opcionales, de modo que aún omitiendo muchas de ellas todavía se podría obtener respuesta del servidor.

Uno de los headers enviados lleva la cookie almacenada en el navegador para el dominio en el que estemos navegando. Puede haber varias cookies separadas por punto y coma, incluyendo un identificador de sesión, por ejemplo:

```
Cookie: PHPSESSID=r2t5uvjq435r4q7ib3vtdjq120; OTRA=mexico
```

En algunos casos las páginas web solicitan usuario y password. En este caso cuando se pulsa en enviar se envía una petición HTTP al servidor donde se incluye un header denominado Authorization que tiene un formato similar a: Authorization: Basic bXl1c2VyOm15cGFzcw==

Donde Basic indica el modo de autorización y la ristra de caracteres es una cadena usuario:password codificada con base64.

CABECERAS DE RESPUESTA (RESPONSE)

Cada petición tiene su respuesta. Una respuesta del servidor consta de una línea de status que contiene alguna información básica sobre la respuesta del servidor, de diversas líneas que constituyen los headers y finalmente una línea en blanco seguida de la respuesta del servidor, normalmente un código HTML extenso. Un ejemplo puede ser:

```
HTTP/1.1 200 OK
Date: Tue, 10 Nov 2015 11:56:29 GMT
Server: Apache/2
X-Powered-By: PHP/5.5.17
P3P: CP="NOI ADM DEV PSAi COM NAV OUR OTRo STP IND DEM"
Expires: Mon, 1 Jan 2001 00:00:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Last-Modified: Tue, 10 Nov 2015 11:56:30 GMT
Vary: Accept-Encoding,User-Agent
Content-Encoding: gzip
Content-Length: 9237
Keep-Alive: timeout=1, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=utf-8

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es-es" lang="es-es" >
  <head>
... continúa un código html extenso
```

La primera línea es la status line o línea de estado, seguida de las cabeceras HTTP y finalmente el contenido servido, normalmente un código HTML.

La línea de estado consta de dos partes: el protocolo, normalmente HTTP 1.1, y un código numérico de estado seguido de un mensaje breve. El código 200 significa que la petición GET tuvo éxito y el servidor devolvió los contenidos solicitados (justo después de las cabeceras).

El típico error 404 proviene del código de error que aparece en la línea de status, 404, que significa not found o no encontrado. En general los códigos 200 indican respuestas correctas, los códigos 300 indican redirecciones, los códigos 400 problemas con la petición y los códigos 500 problemas del servidor.

El resto de cabeceras contienen información sobre la respuesta del servidor, por ejemplo el software que corre el servidor, en qué fecha se hizo la última modificación del recurso solicitado, etc.

Al igual que en las peticiones, muchos de los headers son opcionales de modo que la página podría mostrarse en el navegador incluso si no se recibieran.

En algunos casos dentro de los headers de respuesta del servidor vienen instrucciones para establecer cookies en el navegador, del tipo:

Set-Cookie: skin=noskin; path=/; domain=.amazon.com; expires=Sun, 29-Nov-2009 21:42:28 GMT

Set-Cookie: session-id=120-7333518-8165026; path=/; domain=.amazon.com; expires=Sat Feb 27 08:00:00 2010 GMT

Esto es una forma de establecer cookies, que también pueden ser establecidas de otras maneras (por ejemplo a través de JavaScript).

¿CÓMO PUEDEN VERSE LAS CABECERAS HTTP?

Cuando se mira el código fuente de una página web en un navegador, sólo vemos la porción de respuesta del servidor correspondiente al código HTML, pero no las cabeceras de la respuesta, aunque estas habrán sido transmitidas. Para ver estas cabeceras se debe acudir a herramientas específicas para programadores como la consola web de Firefox, la extensión Firebug de Firefox, la opción Herramientas para desarrolladores – Network de Chrome, etc.

Si usamos algunas de estas herramientas veremos que la carga de una página web implica decenas de peticiones y respuestas, posiblemente una para el html principal, varias para archivos css, varias para archivos javascript, muchas para archivos de imágenes... la transferencia de la información de cada uno de estos elementos lleva aparejada la transmisión de cabeceras, lo que nos da una idea de la complejidad del proceso. Imagen: ejemplo visualizar headers con una herramienta.



ERRORES RELACIONADOS CON LOS HEADERS

Aunque la gestión de headers en cierto grado es realizada por cliente y servidor de forma automática, también puede accederse a su establecimiento o modificación a través de código de programación.

Un error frecuente cuando se trabaja con php es del tipo:

```
Warning: Cannot modify header information - headers already sent by (output started at /some/file.php:12) in /some/file.php on line 23
```

Esto nos indica que los headers no pueden modificarse después de haber sido enviados. Si queremos realizar algún tipo de manipulación en los headers de respuesta del servidor, tiene que ser única y exactamente antes de enviar la respuesta. Si ya hay una respuesta parcialmente enviada, no podemos modificar los headers.

RESUMEN

Además del código HTML que estamos acostumbrados a ver, la transferencia de información entre un cliente y un servidor para la carga de una simple página web implica decenas de peticiones y respuestas (para código, imágenes, archivos javascript, archivos css, etc.) cada una de las cuales contiene cabeceras HTTP (headers) que son líneas de información útiles para la comunicación entre servidor y navegador y viceversa.

Dentro de los headers de petición y respuesta se envía información que resulta útil por ejemplo para establecer o identificar cookies, saber si el navegador acepta envío de información comprimida, etc. Esta información es necesaria para la comunicación, aunque parte de ella es opcional.

EJERCICIO

Usa una herramienta para desarrolladores y visualiza las cabeceras de petición y de respuesta para la carga de la página web <http://aprenderaprogramar.com>. Elige una petición y:

- Muestra las cabeceras de la petición del cliente.
- Muestra las cabeceras de la respuesta del servidor para esa petición.

Para comprobar si tus respuestas son correctas puedes consultar en los foros aprenderaprogramar.com.

Próxima entrega: CU01209F

Acceso al curso completo en aprenderaprogramar.com -- > Cursos, o en la dirección siguiente:
http://aprenderaprogramar.com/index.php?option=com_content&view=category&id=83&Itemid=212