Protocolo HTTP

HTTP, redirecciones, cabeceras, cookies...

HTTP, estándar básico de Internet

- RFC 2616
- http://goo.gl/w7VMj

- Establece:
  - URL
  - Compresión (Gzip, Deflate...)
  - MIME-Types
  - Caché
  - Codificación
  - Métodos (GET, POST, HEAD, PUT...)
  - Encabezados

Métodos de petición de información

- Métodos
  - OPTIONS: posibilidades del servidor (tamaños, codificación, etc...)
  - GET: Petición por URI
  - HEAD: Igual que un GET, sin contenido
  - POST: Información «enclosed»
  - PUT: POST especial (URI identifica al elemento)
  - DELETE: Elimina la URI
  - TRACE: Hace seguimiento del mensaje por «loop-back»
  - CONNECT: Uso en proxy que cambia a un tunnel (SSL)
- ¿Eliminar las cabeceras no útiles?

Eliminar cabeceras no utilizables

```
<Limit POST PUT DELETE CONNECT>
# Require valid-user
Deny from all
</Limit>
```

- Protección frente ataques
- Limitamos las peticiones a elementos que la web sí utiliza, como POST o GET.

Códigos de estado

- **1xx - Informational**: información recibida o proceso continuado.
- **2xx - Success**: La acción se ha recibido correctamente, se ha entendido y aceptado.
- **3xx - Redirection**: Se requiere de una acción apara acabar la petición.
- **4xx - Client Error**: La petición contiene errores de sintaxis o no se puede llevar a cabo.
- **5xx - Server Error**: El servidor generó un error contestando una petición aparentemente correcta.

Códigos de estado

- Los 100 son normales (no se deberían ver nunca), los 200 son correctos, los 300 son redirecciones, los 400 son errores «leves» y los 500 son errores «graves».
- Lo mejor es que sólo haya códigos 200 (OK) o 304 (Not Modified).
- Los códigos 400 hay que intentar reducirlos
- Los códigos 500 hay que evitarlos y/o corregirlos inmediatamente.

Códigos de estado

- 100: Continue
- 101: Switching Protocols
- 200: OK
- 201: Created
- 202: Accepted
- 203: Non-Authoritative Information
- 204: No Content
- 205: Reset Content
- 206: Partial Content
- 300: Multiple Choices
- 301: Moved Permanently
- 302: Found
- 303: See Other
- 304: Not Modified
- 305: Use Proxy
- 307: Temporary Redirect

Códigos de estado

- 400: Bad Request
- 401: Unauthorized
- 402: Payment Required
- 403: Forbidden
- 404: Not Found
- 405: Method Not Allowed
- 406: Not Acceptable
- 407: Proxy Authentication Required
- 408: Request Time-out
- 409: Conflict
- 410: Gone
- 411: Length Required
- 412: Precondition Failed
- 413: Request Entity Too Large
- 414: Request-URI Too Large
- 415: Unsupported Media Type
- 416: Requested range not satisfiable
- 417: Expectation Failed |

Códigos de estado

- 500: Internal Server Error
- 501: Not Implemented
- 502: Bad Gateway
- 503: Service Unavailable
- 504: Gateway Time-out
- 505: HTTP Version not supported

- ¿Pero cuáles de estos son los más habituales, los que más se escuchan y los que realmente tienen cierto sentido de uso?

Códigos 200

- *200 OK* - Este es el código habitual cuando se devuelve una página. Es el “normal”, así que poco a decir tengo.
- *204 No Content* - Este código lo vi usar hace tiempo en Google. No recuerdo exactamente cuándo, pero creo que es la forma en la que hacen las redirecciones a otras páginas desde sus resultados de búsqueda. El servidor devuelve una cabecera pero no devuelve “contenido”.

Códigos 300

- *301 Moved Permanently* - Poco hay que decir de este caso. Es la forma en la que si tienes una página que «ya no existe» y tienes otra que la sustituye, con esto consigues que tanto usuario como buscador vayan a la nueva. En el caso del buscador «elimina» la URL vieja e indexa esta «nueva» URL.
- *302 Found* - ¡Coño! ¿Found? ¿no era Redirect Temporaly? Pues sí, aquí está el secreto de porqué los buscadores «indexan» los 302 y es que, aunque se hace una redirección, la respuesta es «Found», o sea, que se ha encontrado y, por tanto, como la página se ha encontrado, se «toma prestado» el contenido de la página a la que se redirecciona y se “pone” también en la URL antigua. De ahí que genere contenidos duplicados un 302.

Códigos 300

- *304 Not Modified* - Este también es un clásico, sobretodo cuando hablamos de contenidos cacheados.
- *307 Temporary Redirect* - ¡Anda! Si no es 302, ¡es 307! Sí, si lo que queremos es «distribuir» de forma aleatoria visitas, hay que utilizar un 307. Con este código (que personalmente no he probado con buscadores) se supone que la página original no se debería indexar, ni ella ni la de destino, ya que “cada vez” puede cambiar. Supongo que la de destino, si el buscador es listo, la indexará como una URL nueva, sin tener que ver de dónde viene.

Códigos 400/500

- *403 Forbidden* - El servidor entiende lo que se le está pidiendo, pero no quiere contestar nada. Es como un 404, pero «educado».
- *404 Not Found* - Si no encuentro nada que hacer y no sé si lo volveré a encontrar, te contesto esto, pero que sepas que no tengo ni idea ni de porqué te estoy contestando esto. Es decir, que devolver un 404 y no devolver nada es casi lo mismo… el 404 es para administradores y programadores «vagos» que no controlan su sitio web.

Códigos 400/500

- *410 Gone* - La frase en inglés lo deja muy claro: «The requested resource is no longer available at the server and no forwarding address is known». Esto significa que lo que había aquí ya no está y no se sabe dónde está; traducido al idioma de los buscadores: «si tenías esto indexado, bórralo porque no existe y si existiera no sé donde está». Si hay que borrar algo de un buscador, es la respuesta perfecta.
- *503 Service Unavailable* - Cuando hay una sobrecarga en la web hay que devolver este código. También significa que el servidor está en mantenimiento; esto significa que, si en algún momento hay que poner un mensaje de «Página en Mantenimiento» ha de ir acompañada de este código. Por cierto, este código siempre ha de ir acompañado de un «Retry-After» porque sino se comporta como un «500 Internal Server Error».

Tipos de caché

- *public*: puede cachearse en cualquier momento y lugar (habitualmente la caché del navegador).
- *private*: puede ser cacheado pero en una caché que sólo pueda leer el usuario que está accediendo en ese momento (no en una caché compartida).
- *no-cache*: no debe ser cacheado el documento.
- *no-store*: cuando una página se carga, se suele hacer una copia en el disco (ya esté cacheada o no)… con esto impides que se haga dicha copia.
- *s-maxage*: en cachés públicas, esta cabecera devuelve la edad máxima de la caché; además, si hay un proxy por algún lado, le obligará a no cachear.
- *max-age*: la edad de la página no puede ser mayor que la especificada en segundos.
- *max-stale*: se puede recibir información pero no después de los segundos que se indican.
- *min-fresh*: la página ha de refrescarse en no menos que el momento actual sumado los segundos.
- *must-revalidate*: aunque en algún lado se indique que hay que cachear, con esto forzamos a que las comunicaciones se hagan sin cacheos.

Redirecciones ¿cuándo y cómo?

- Lo mejor es que las URL no cambien nunca
- Si han de cambiarse a una página que existe, que devuelva un 301.
- Si han de cambiarse a algo que no existe, que devuelva un 410.
- En ambos casos, el tiempo recomendado de uso de estos sistemas es de 6 meses.
- Los códigos 301 deben ir seguidos de una cabecera Expires que indique «la caché» de esa redirección. Debería ser de 3 o 6 meses.

Redirecciones ¿cuándo y cómo?

```
$url_redir = "Location:
http://example.com/page.html";
header($url_redir, true, 301);
$expires = "Expires: ".date("D, d M Y
H:i:s T", time()+86400);
header($expires);
exit;
```

- Calculamos la URL de destino
- Enviamos la cabecera con la redirección 301
- Calculamos la fecha de caducidad

Analizando las cabeceras

- Petición:

```
http://www.javiercasares.com/
```

Analizando las cabeceras

- Información:

```
GET / HTTP/1.1
Host: www.javiercasares.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.2.17) Gecko/20110420 Firefox/3.6.17
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en,es;q=0.7,ca;q=0.3
Accept-Encoding: gzip,deflate
Accept-Charset: UTF-8,*
Keep-Alive: 115
Connection: keep-alive
Cookie: __utma=1304599264.105;
__utmz=1303186555.1.1.utmcsr=(direct)|utmccn=(direct)|utmcmd=(none)
```

Analizando las cabeceras

- Respuesta:

```
HTTP/1.1 301 Moved Permanently
Date: Mon, 30 May 2011 12:44:26 GMT
Server: Apache/2.2.3 (CentOS)
Location: http://javiercasares.com/
Content-Length: 233
Connection: close
Content-Type: text/html; charset=iso-8859-1
```

Analizando las cabeceras

- Petición:

```
http://javiercasares.com/
```

Analizando las cabeceras

- Información:

```
GET / HTTP/1.1
Host: javiercasares.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US;
rv:1.9.2.17) Gecko/20110420 Firefox/3.6.17
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en,es;q=0.7,ca;q=0.3
Accept-Encoding: gzip,deflate
Accept-Charset: UTF-8,*
Keep-Alive: 115
Connection: keep-alive
Cookie: wp-settings-
1=m0%3Dc%26m1%3Dc%26m2%3Dc%26m3%3Dc%26m4%3Dc%26m5%3Dc%26m6%3Dc%26m7%3D
c%26m8%3Dc%26m9%3Dc%26m10%3Dc%26m11%3Dc%26m12%3Dc%26align%3Dcenter%26u
rlbutton%3Dfile%26imgsize%3Dlarge%26m13%3Dc%26m14%3Dc%26m15%3Dc%26m16%
3Dc%26m17%3Dc%26m18%3Dc%26m19%3Dc%26m20%3Dc%26m21%3Dc%26m22%3Dc%26m23%
3Dc%26m24%3Do%26m25%3Dc%26wplink%3D1%26galfile%3D1; wp-settings-time-
1=1306658887;
__utma=119895699.710930638.1303186555.1304598560.1304599264.105;
__utmz=119895699.1303186555.1.1.utmcsr=(direct)|utmccn=(direct)|utmcmd
=(none)
```

Analizando las cabeceras

- Respuesta:

```
HTTP/1.1 200 OK
Date: Mon, 30 May 2011 12:44:26 GMT
Server: Apache/2.2.3 (CentOS)
X-Powered-By: PHP/5.2.10
X-Pingback: http://javiercasares.com/xmlrpc.php
Last-Modified: Mon, 30 May 2011 09:28:24 GMT
Content-Encoding: gzip
Vary: Accept-Encoding
Connection: close
Transfer-Encoding: chunked
Content-Type: text/html; charset=UTF-8
```

Dominios sin cookies

- Los contenidos no programables no necesitan recibir peticiones de cookies o información «absurda».
- Los dominios sin cookies no reciben (ni crean) información de cookies. Tampoco necesitan información suplementaria.
- Los dominios sin cookies han de estar en un dominio separado, no sirve un subdominio. Algunos navegadores transfieren las cookies del dominio a sus subdominios de forma automática.

Dominios sin cookies

```
<VirtualHost *:80>
# asignamos la carpeta
DocumentRoot /data/
# asignamos el dominio
ServerName domstatic.com
# eliminamos las cookies
RequestHeader unset Set-Cookie
Header unset Set-Cookie
# asignamos el etag sin control del "servidor" (lo que permite
escalar en varios frontales)
FileEtag MTime Size
# asignamos compresión por defecto a ALGUNOS contenidos
DeflateCompressionLevel 9
        <FilesMatch "\.(js|css|txt|htm|html)$">
        SetOutputFilter DEFLATE
        BrowserMatch ^Mozilla/4 gzip-only-text/html
        BrowserMatch ^Mozilla/4\.0[678] no-gzip
        BrowserMatch \bMSIE !no-gzip !gzip-only-text/html
        </FilesMatch>
</VirtualHost>
```

Dominios sin cookies

- Petición «con cookies»:

```
GET /multimedia/imagen.jpg HTTP/1.1
Host: www.example.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US;
rv:1.9.2.17) Gecko/20110420 Firefox/3.6.17
Accept: image/png,image/*;q=0.8,*/*;q=0.5
Accept-Language: en,es;q=0.7,ca;q=0.3
Accept-Encoding: gzip,deflate
Accept-Charset: UTF-8,*
Keep-Alive: 115
Connection: keep-alive
Referer: http://www.example.com/
Cookie:
__utma=269635316.1628702604.1304409938.1304494842.1304666667.
3;
__utmz=269635316.1304409938.1.1.utmcsr=(direct)|utmccn=(direc
t)|utmcmd=(none); PHPSESSID=6ii2ppk6qoii8tpi2gfv17pho3
```

Dominios sin cookies

- Respuesta:

```
HTTP/1.1 200 OK
Date: Mon, 30 May 2011 12:53:34 GMT
Server: Apache/2.2.3 (CentOS)
Last-Modified: Fri, 06 May 2011 10:07:35 GMT
Accept-Ranges: bytes
Content-Length: 178078
Cache-Control: max-age=315360000
Expires: Thu, 27 May 2021 12:53:34 GMT
Connection: close
Content-Type: image/jpeg
```

Dominios sin cookies

- Petición «sin cookies»:

```
GET /multimedia/imagen.jpg HTTP/1.1
Host: domstatic.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1;
en-US; rv:1.9.2.17) Gecko/20110420 Firefox/3.6.17
Accept:
text/html,application/xhtml+xml,application/xml;q=0.
9,*/*;q=0.8
Accept-Language: en,es;q=0.7,ca;q=0.3
Accept-Encoding: gzip,deflate
Accept-Charset: UTF-8,*
Keep-Alive: 115
Connection: keep-alive
```

Dominios sin cookies

- Respuesta:

```
HTTP/1.1 200 OK
Date: Mon, 30 May 2011 12:56:12 GMT
Server: Apache/2.2.3 (CentOS)
Last-Modified: Mon, 23 May 2011 09:49:53 GMT
Etag: "2b79e-4a3ee64a63240"
Accept-Ranges: bytes
Content-Length: 178078
Connection: close
Content-Type: image/jpeg
```

Gracias

Javier Casares
javier.casares@kisslab.com - http://javiercasares.com