



JAVA A PROFUNDIDAD



Códigos HTTP

El formato de las peticiones HTTP request y HTTP response son similares y ambos tendrán la siguiente estructura.

- Una línea de status inicial + CRLF (Retorno de carro + Línea de espera. Por ejemplo, una nueva línea).
- Cero o más líneas de cabecera + CRLF.
- Una línea en blanco.
- Un cuerpo opcional de mensaje como un archivo, datos de consulta, o salida de consulta.

Por ejemplo, un servidor responde cabeceras como las siguientes:

```
HTTP/1.1 200 OK
Content-Type: text/html
Header2: ...
...
HeaderN: ...
(Blank Line)
<!doctype ...>
<html>
  <head>...</head>
  <body>
    ...
  </body>
</html>
```

La línea de status consiste de la versión de HTTP (HTTP/1.1 en el ejemplo), un código de status (200 en el ejemplo), y un mensaje corto que corresponde al código de status (OK en el ejemplo).

La siguiente es una lista de los códigos de status HTTP y sus mensajes asociados que pueden ser devueltos desde el servidor web.

Código	Mensaje	Descripción
100	Continue	Solo una parte de la petición ha sido recibida por el servidor, pero mientras no haya sido rechazada, el cliente debe continuar con la solicitud.
101	Switching Protocols	El servidor cambia de protocolo.
200	OK	La petición es OK.
201	Created	La petición está completa, y un nuevo recurso ha sido creado.
202	Accepted	La petición es aceptada para procesamiento, pero el procesamiento no está completo.
203	Non-authoritative Information	
204	No content	
205	Reset content	

Sitio web: www.kodikas.com.mx

Facebook: <https://www.facebook.com/profeJavierV/>

YouTube: <https://www.youtube.com/user/jvazquezolivares>



JAVA A PROFUNDIDAD



206	Partial Content	
300	Multiple Choices	Una lista de enlaces. El usuario puede seleccionar un enlace e ir a esa ubicación. Máximo cinco direcciones.
301	Moved permanently	La página solicitada se ha movido a una nueva url.
302	Found	La página solicitada se ha movido temporalmente a una nueva url.
303	See Other	La página solicitada puede ser encontrada en una url diferente.
304	Not Modified	
305	Use Proxy	
307	Temporary Redirect	La página solicitada ha sido movida temporalmente a una nueva url.
400	Bad Request	El servidor no entendió la petición.
401	Unauthorized	La página solicitada necesita un usuario y un password.
402	Payment Required	No es posible usar este código aún.
403	Forbidden	El acceso a la página solicitada está prohibido.
404	Not Found	El servidor no puede encontrar la página solicitada.
405	Method Not Allowed	El método especificado en la petición no está permitido.
406	Not Acceptable	El servidor solo puede generar una respuesta que no sea aceptada por el cliente.
407	Proxy Authentication Required	Debes autenticarte con un servidor proxy antes de que esta petición pueda ser servida.
408	Request Timeout	La petición tomo más tiempo de lo que el servidor estaba preparado para esperar.
409	Conflict	La petición no puede ser completada a causa de un conflicto.
410	Gone	La página solicitada no está disponible ya.
411	Length Required	El "Content-Length" no está definido. El servidor no aceptará la petición sin esta cabecera.
412	Precondition Failed	La precondition dada en la petición ha sido evaluada como falsa por el servidor.
413	Request Entity Too Large	El servidor no aceptará la petición porque la entidad de petición es demasiado grande.
414	Request-url Too Long	El servidor no aceptará la petición, porque la url es demasiado larga. Ocurre cuando convertimos una petición post a una petición get con una cadena de información demasiado larga.
415	Unsupported Media Type	El servidor no aceptará la petición, porque el tipo de archivo multimedia no está soportado.

Sitio web: www.kodikas.com.mx

Facebook: <https://www.facebook.com/profeJavierV/>

YouTube: <https://www.youtube.com/user/jvazquezolivares>



JAVA A PROFUNDIDAD



417	Expectation Failed	
500	Internal Server Error	La petición no fue completada. El servidor encontró una condición inesperada.
501	Not Implemented	La petición no fue completada. El servidor no soportó la funcionalidad requerida.
502	Bad Gateway	La petición no fue completada. El servidor recibió una respuesta inválida desde el upstream server.
503	Service Unavailable	La petición no fue completada. El servidor está temporalmente fuera de servicio o caído.
504	Gateway Timeout	Se ha terminado el tiempo del Gateway
505	HTTP Version Not Supported	El servidor no soporta la versión de protocolo http utilizada.

Métodos para establecer código de status HTTP.

Los siguientes métodos pueden ser usados para establecer código de status HTTP en nuestros servlets. Estos métodos están disponibles con los objetos HttpServletResponse.

public void setStatus(int statusCode)

Este método establece un código de status arbitrario. El método setStatus toma un int (el código de status) como un argumento. Si tu respuesta al cliente incluye un código de status especial y un documento, asegúrate de llamar a setStatus antes de devolver realmente el contenido con el objeto PrintWriter.

public void sendRedirect(String url)

Este método genera una respuesta 302 junto con un encabezado de ubicación que proporciona la URL del nuevo documento.

Código de ejemplo

El siguiente es un ejemplo en el cual enviamos un código de error 407 al cliente y el browser del cliente mostrará el mensaje "Necesitas autenticación".

```
1. import java.io.*;
2. import javax.servlet.*;
3. import javax.servlet.http.*;
4. import java.util.*;
5.
6. public class showError extends HttpServlet {
7.
8.     // Método que maneja las peticiones por el método GET.
9.     public void doGet(HttpServletRequest request, HttpServletResponse response)
```

Sitio web: www.kodikas.com.mx

Facebook: <https://www.facebook.com/profeJavierV/>

YouTube: <https://www.youtube.com/user/jvazquezolivares>



JAVA A PROFUNDIDAD



```
10.     throws ServletException, IOException {
11.
12.     // Establece un código de error y una razón del error.
13.     response.sendError(407, "Necesita autenticación!!!" );
14. }
15.
16. // Método que maneja las peticiones por el método POST.
17. public void doPost(HttpServletRequest request, HttpServletResponse response)
18.     throws ServletException, IOException {
19.
20.     doGet(request, response);
21. }
22. }
```

