SISTEMAS WEB CURSO 2022/2023

HTTP - HyperText Transfer Protocol Compresión



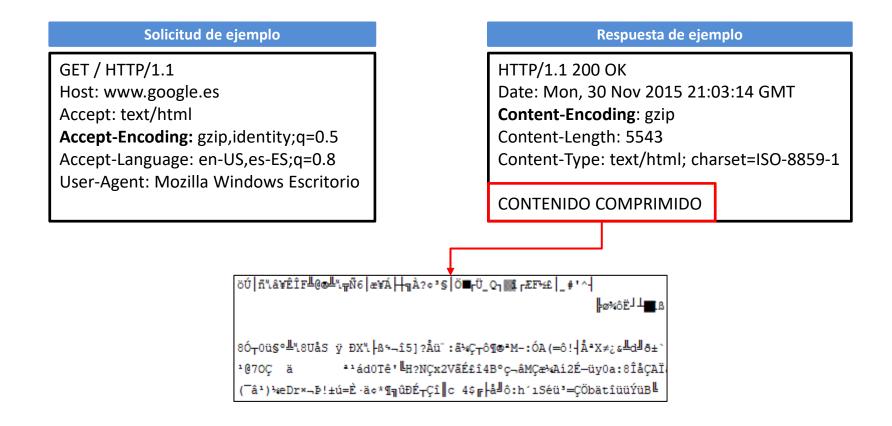
Web Sistemak by <u>Oskar Casquero</u> & <u>María Luz Álvarez</u> is licensed under a <u>Creative Commons Reconocimiento 4.0 Internacional License</u>.

FUNCIONAMIENTO DE HTTP: COMPRESIÓN

- ¿Cómo se reduce el tamaño de las respuestas?
 - —En general, el contenido de una respuesta HTTP es texto: HTML, XML, JSON, CSS.
 - El texto tiene buena respuesta a la compresión.
 - -El protocolo HTTP permite comprimir el cuerpo del mensaje, siendo el algoritmo más utilizado para ello el gzip.
 - —El algoritmo de compresión se indica en dos cabeceras:
 - En la solicitud: "Accept-Encoding"
 - Si el cliente no tiene capacidad para descomprimir un contenido comprimido, este encabezamiento tomará el valor de "Identity".
 - En la respuesta: "Content-Encoding"
 - Si el cuerpo del mensaje se envía sin comprimir, esta cabecera no se debe introducir en la respuesta.

FUNCIONAMIENTO DE HTTP: COMPRESIÓN

 Supongamos que un cliente, capaz de trabajar con contenidos comprimidos, solicita un recurso con URI: http://www.google.es/



FUNCIONAMIENTO DE HTTP: COMPRESIÓN - EJEMPLO

```
import requests
import sys
import zlib
metodo = 'GET'
                                               • La libreria requests descomprime automáticamente los
uri = "https://www.google.es/"
cabeceras = {'Host': 'www.google.es'}
                                                 formatos de compresión gzip y deflate.
compressed = False
                                               • Si gueremos un acceso directo al contenido de la
if len(sys.argv) == 1:
                                                 respuesta, introducimos el parámetro stream = True en la
    cabeceras['Accept-Encoding'] = 'identity'
elif sys.argv[1] == 'compress':
                                                 solicitud y leemos el campo raw.data en la respuesta
    compressed = True
    cabeceras['Accept-Encoding'] = 'qzip'
else:
   print("Error! Erabilera: python compression es.py compress")
    exit(0)
respuesta = requests.request (metodo, uri, headers=cabeceras, allow redirects=False, stream=True
codigo = respuesta.status code
descripcion = respuesta.reason
print(str(codigo) + " " + descripcion)
for cabecera in respuesta.headers:
    print(cabecera + ": " + respuesta.headers[cabecera])
print("RESPONSE CONTENT LENGTH: " + str(len(respuesta.raw.data)) + " byte")
if compressed:
    contenido compressed = respuesta.raw.data
    contenido uncompressed = zlib.decompress(contenido compressed, 16+zlib.MAX WBITS)
    print("UNCOMPRESSED RESPONSE CONTENT LENGTH: " + str(len(contenido uncompressed)) + " byte")
```

FUNCIONAMIENTO DE HTTP: COMPRESIÓN - EJEMPLO

```
(venv) C:\Users\cvzcaoio\Dropbox (Personal)\docencia\grado\Sistemas Web\SW 2020\Praktikal>python compression.py compress
 200 OK
 Date: Wed, 12 Feb 2020 07:39:19 GMT
 Expires: -1
 Cache-Control: private, max-age=0
 Content-Type: text/html; charset=ISO-8859-1
 P3P: CP="This is not a P3P policy! See g.co/p3phelp for more info."
 Content-Encoding: gzip
 Server: gws
X-XSS-Protection: 0
X-Frame-Options: SAMEORIGIN
 Set-Cookie: 1P JAR=2020-02-12-07; expires=Fri, 13-Mar-2020 07:39:19 GMT; path=/; domain=.google.es; Secure, NID=197=TuCqh
 sRxyTUWCZOVawM6ScZObD0; expires=Thu, 13-Aug-2020 07:39:19 GMT; path=/; domain=.google.es; HttpOnly
 Alt-Svc: guic=":443"; ma=2592000; v="46,43",h3-0050=":443"; ma=2592000,h3-0049=":443"; ma=2592000,h3-0048=":443"; ma=2592000,h3-0048=":443"; ma=2592000,h3-0048=":443"; ma=2592000,h3-0049=":443"; ma=2592000,h3-0049=":443"
Transfer-Encoding: chunked
 RESPONSE CONTENT LENGTH: 5584 byte
 UNCOMPRESSED RESPONSE CONTENT LENGTH: 13116 byte
```

- Tamaño de la respuesta comprimida: 5584 octetos
- Tamaño de la respuesta sin comprimir: 13116 octetos
 Un 57% de ahorro en el tamaño del cuerpo del mensaje