All About HTTP Chunked Responses

Gilly Barr

4 Sep 2013 CPOL

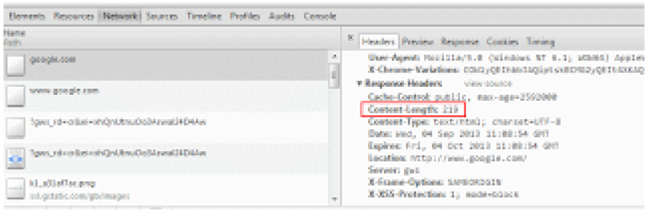
Rate me: ★★★★★ 5.00/5 (1 vote)

This post will discuss all about HTTP chunked responses

A Short Background on HTTP and the 'Content-Length' Header

When sending requests over HTTP (hence, 'the web'), we send an HTTP request which consists of two main parts - the header of the request and the body. The header defines various details of the request body (e.g.: encoding type, cookies, request method, etc.). One of these details is the '`Content-Length`' specifying the size of the body. If you're building a website and aren't specifying this explicitly, then chances are the framework you're using is doing this for you. Once you send the response to the client, the framework measures the size of the response and adds it to this header.

In a normal request, looking at the headers with FireBug or Chrome developer tools, it should look like this (looking at google.com):


Elements Resources Network Sources Timeline Profiles Audits Console
Name
Path
google.com
www.google.com
k1_a31af7ac.png
ssl.gstatic.com/gb/images
Headers Preview Response Cookies Timing

```
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) Apple
X-Chrome-Variations: [illegible]
▼ Response Headers    view source
Cache-Control: public, max-age=2592000
Content-Length: 219
Content-Type: text/html; charset=UTF-8
Date: Wed, 04 Sep 2013 11:08:54 GMT
Expires: Fri, 04 Oct 2013 11:08:54 GMT
Location: http://www.google.com/
Server: gws
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
```

So, What is a 'Chunked Response' ?

A 'chunked' response means that instead of processing the whole page, generating all of the HTML and sending it to the client, we can split the HTML into 'chunks' and send one after the other, without telling the browser how big the response will be ahead of time.

Why Would Anyone Want To Do This ?

Well, some pages on the site can take a long time to process. While the server is working hard to generate the output, the user sees a white screen and the browser is pretty much hopeless during this time with nothing to do and just displays a boring white screen to the user.

The work the server is doing might be to generate a specific part of the content on the page, and we might have a lot ready that we can already give the client to work with. If you have scripts & stylesheets in the `<head/>` of your page, you can send the first chunk with the '`head`' tag HTML content to the user's machine. Then the browser will have something to work with, meaning it will start downloading the scripts and resources it needs and during this time, your servers can continue crunching numbers to generate the content to be displayed.

You are actually gaining parallelism by sending the client this first chunk without waiting for the rest of the page to be ready!

Taking this further, you can split the page into several chunks. In practice, you can send one chunk with the '`head`' of the page. The browser can then start downloading scripts and stylesheets, while your server is processing let's say the categories from your db to display in your header menu/navigation. Then you can send this as a chunk to the browser so it will have something to start rendering on the screen, and your server can continue processing the rest of the page.

Even if the user only sees part of the content, and it isn't enough to work with, the user still gets a 'sense' of better performance - something we call 'perceived performance' which has almost the same impact.

Many big sites are doing this, since this will most definitely improve the client side performance of your site. Even if it's only by a few milliseconds, in the ecommerce world, we know that time is money!

How Does This Work?

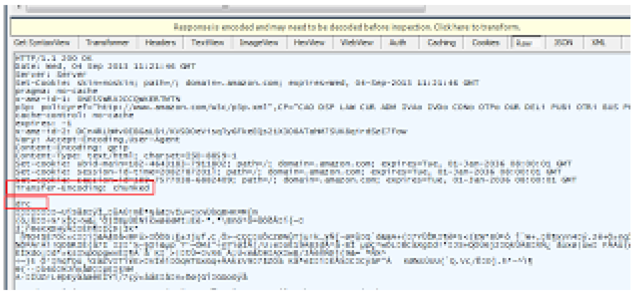
Since the response is chunked, you cannot send the '`Content-Length`' response header because you don't necessarily know how long the response will be. Usually, you won't know how big the response will be, and even if you do, the browser doesn't care at this point.

So, to notify the browser about the chunked response, you need to omit the '`Content-Length`' header, and add the header '`Transfer-Encoding: chunked`'. Giving this information to the browser, the browser will now expect to receive the chunks in a very specific format.

At the beginning of each chunk, you need to add the length of the current chunk in hexadecimal format, followed by '`\r\n`' and then the chunk itself, followed by another '`\r\n`'.

FireBug and Chrome dev tools both combine the chunks for you, so you won't be able to see them as they are really received by the browser. In order to see this properly, you will need to use a more low level tool like Fiddler.

This is how the raw response of amazon.com looks like using Fiddler:



```
HTTP/1.1 200 OK
Date: Wed, 04 Sep 2013 11:21:46 GMT
Server: Server
Set-Cookie: [illegible]; path=/; domain=.amazon.com; expires=Wed, 04-Sep-2033 11:21:46 GMT
pragma: no-cache
x-amz-id-1: [illegible]
p3p: policyref="http://www.amazon.com/w3c/p3p.xml",CP="CAO DSP LAW CUR ADM IVAo IVDo CONo OTPo OUR DELi PUBi OTRi BUS PHY
cache-control: no-cache
expires: -1
x-amz-id-2: [illegible]
Vary: Accept-Encoding,User-Agent
Content-Encoding: gzip
Content-Type: text/html; charset=ISO-8859-1
Set-cookie: ubid-main=[illegible]; path=/; domain=.amazon.com; expires=Tue, 01-Jan-2036 08:00:01 GMT
Set-cookie: session-id-time=[illegible]; path=/; domain=.amazon.com; expires=Tue, 01-Jan-2036 08:00:01 GMT
Set-cookie: session-id=[illegible]; path=/; domain=.amazon.com; expires=Tue, 01-Jan-2036 08:00:01 GMT
Transfer-Encoding: chunked

d7c
[illegible]
```

Note: I marked the required '`Transfer-Encoding: chunked`' header, and the first line with the size of the chunk. In this case, the first chunk is 0xd7c bytes long, which in human-readable format is 3452 bytes.

Also, it's interesting to note that you cannot really read the first chunk since it's encoded via gzip (which is also automatically decoded when using browser dev tools). When using Fiddler, you can see the message at the top telling you this, and you can click it and have it decoded, but then the chunks are removed and you'll see the whole HTML output.

How Can We Achieve This With ASP.NET?

When you want to flush the content of your site, all you need to do in the middle of a view is call '`HttpContext.Current.Response.Flush()`'.

It's that easy! Without you having to worry about it, the .NET Framework will take care of the details and send the response to the browser in the correct format.

Some things that might interfere with this working properly are as follows:

- You might have to configure '`Response.BufferOutput = false;`' at the beginning of your request so the output won't be buffered and will be flushed as you call it.
- If you specifically add the '`Content-Length`' header yourself, then this won't work.

For More Helpful Resources on Chunked Responses

- Wikipedia, and the spec details: http://en.wikipedia.org/wiki/Chunked_transfer_encoding

- How to write chunked responses in .NET (but not ASP.NET) - http://blogs.msdn.com/b/asiatech/archive/2011/04/26/how-to-write-chunked-transfer-encoding-web-response.aspx
- Implementing chunked with `IHttpListener` - http://www.differentpla.net/content/2012/07/streaming-http-responses-net

License

Share

About the Author

Gilly Barr
Web Developer
Israel

Started programming e-commerce sites with PHP & MySQL at the age of 14. Worked for me well for about 5 years.

Transfered to C# & asp.net, while serving in the IDF.
Worked on the 'Core Performance' Team at ShopYourWay.com (Sears Israel)
Currently working at Logz.io

Check out my blog!
or my twitter

Comments and Discussions

You must Sign In to use this message board.

Search Comments

First Prev Next

My vote of 5

Mahesh Babu Kudikala **4-Sep-13 7:59**

Re: My vote of 5
Martinux-net 6-Sep-13 2:24

Refresh 1

General News Suggestion Question Bug Answer Joke Praise Rant Admin

Use Ctrl+Left/Right to switch messages, Ctrl+Up/Down to switch threads, Ctrl+Shift+Left/Right to switch pages.

Permalink
Advertise
Privacy
Cookies
Terms of Use

Layout: fixed | fluid

Web04 2.8.20210412.1