



( / )

# HTTP Gallery

---

## 9. Content Length & Transfer Encoding

When an HTTP client is reading a response message from a server it needs to know when it has reached the end of the message. This is particularly important with persistent (keep alive) connections, because a connection can only be re-used by another HTTP transaction after the response message has been fully received. The following sections describe the four ways in which an HTTP server can indicate the end of the response message:

### 9.1 Connection Closed by Server

The connection can be closed at the end of the response message by the server, but this prevents connections being re-used.

### 9.2 Content-Length Header

The length of the content after the response headers can be specified in bytes with the **Content-Length** header

### 9.3 Implied Content Length

Some types of responses, such as **304**, are defined to never have content and therefore the client can assume that the response message is terminated by the double CRLF after the headers.

## 9.4 Chunked Encoding

The content can be broken up into a number of chunks; each of which is prefixed by its size in bytes. A zero size chunk indicates the end of the response message. If a server is using chunked encoding it must set the **Transfer-Encoding** header to "chunked".

Chunked encoding is useful when a large amount of data is being returned to the client and the total size of the response may not be known until the request has been fully processed. An example of this is generating an HTML table of results from a database query. If you wanted to use the **Content-Length** header you would have to buffer the whole result set before calculating the total content size. However, with chunked encoding you could just write the data one row at a time and write a zero sized chunk when the end of the query was reached.

### Example 9

This image (chunkedimage.aspx) has been supplied\* to your browser using a 1 KB chunk of data every 0.1 seconds:



**REFRESH THIS IMAGE**

(\* if you are accessing this page through a proxy it may remove the chunked encoding)

## Using HttpWatch with Example 9

To view the chunked response discussed on this page:

1. Open HttpWatch by right clicking on the web page and selecting HttpWatch from the context menu
2. Click on **Record** to start logging requests in HttpWatch
3. Click on the **Refresh** button above
4. Select the entry for this HTML page and go to the **Streams** tab
5. The number of chunks is displayed in this tab and the format of the length prefixed chunks can be seen after the response headers.

< **8. COMPRESSION** (../COMPRESSION/)

**10. AUTHENTICATION** > (../AUTHENTICATION/)

Ready to get started?

**TRY FOR FREE**

**(/DOWNLOAD/)  
BUY NOW (/BUY/)**

---

HttpWatch

Features (/features/httpdebugger.aspx)

Compare Editions (/editions.aspx)

[New in Version 11.x \(/newfeatures.aspx\)](#)

[Download \(/download/\)](#)

[Pricing \(/buy/\)](#)

## Our Customers

[Who is using it? \(/#customers\)](#)

[What are they saying? \(/#quotes\)](#)

## Learning & Documentation

[HttpWatch Blog \(https://blog.httpwatch.com\)](https://blog.httpwatch.com)

[HTTP Gallery \(/httpgallery/\)](#)

[HttpWatch Help \(https://help.httpwatch.com\)](https://help.httpwatch.com)

[HttpWatch Automation Reference \(https://apihelp.httpwatch.com\)](https://apihelp.httpwatch.com)

## Support

[Technical Support \(/support/\)](#)

[About Us \(/company/\)](#)

[Contact Us \(/company/\)](#)

[Blog \(https://blog.httpwatch.com\)](https://blog.httpwatch.com)

[Twitter \(https://twitter.com/httpwatch\)](https://twitter.com/httpwatch)

---

[Search \(/search/\)](#)   [Terms & Conditions \(/company/terms.aspx\)](#)   [Privacy Policy \(/company/privacy\\_policy.aspx\)](#)

[Copyright \(/company/copyright.aspx\)](#)

Copyright © 2019 Neumetrix Limited

