
Subject: [SOLVED]HttpRequest will hang at ReadingBody() until timeout When KeepAlive is set

Posted by [kasome](#) on Mon, 30 Apr 2018 12:50:10 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi,

Can HttpRequest get Content-Length=0 from response header instead of reading EOF from the server to confirm the requested operation is done When KeepAlive is set to true ?

From RFC2616

Persistent HTTP connections have a number of advantages :

Quote:

- By opening and closing fewer TCP connections, CPU time is saved in routers and hosts (clients, servers, proxies, gateways, tunnels, or caches), and memory used for TCP protocol control blocks can be saved in hosts.
- HTTP requests and responses can be pipelined on a connection. Pipelining allows a client to make multiple requests without waiting for each response, allowing a single TCP connection to be used much more efficiently, with much lower elapsed time.
- Network congestion is reduced by reducing the number of packets caused by TCP opens, and by allowing TCP sufficient time to determine the congestion state of the network.
- Latency on subsequent requests is reduced since there is no time spent in TCP's connection opening handshake.
- HTTP can evolve more gracefully, since errors can be reported without the penalty of closing the TCP connection. Clients using future versions of HTTP might optimistically try a new feature, but if communicating with an older server, retry with old semantics after an error is reported.

any suggest will be welcome, thanks in advance

Subject: Re: [FeatureRequests]HttpRequest will hang at ReadingBody() until timeout When KeepAlive is set

Posted by [mirek](#) on Tue, 08 May 2018 12:28:56 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hopefully fixed, please check.

Mirek

Subject: Re: [FeatureRequests]HttpRequest will hang at ReadingBody() until
timeout When KeepAlive is set

Posted by [kasome](#) on Fri, 11 May 2018 06:14:23 GMT

[View Forum Message](#) <> [Reply to Message](#)

Works like a charm . Thanks a lot.
