

Sensible HTTP POST timeout values to use when programmatically issuing requests?

Asked 16 years, 2 months ago Modified 16 years, 2 months ago

Viewed 15k times



When programmatically issuing HTTP POST requests, what timeout values would be sensible?

10



In my case, I'm looking to set 'sensible' timeout values when making POST requests in PHP, however this applies to any language.



I need to be able to issue a set of requests, each to a user-specified URL. If I do need to process requests consecutively instead of concurrently, I'd like to specify a sensible time beyond which a request is deemed to have timed out.

PHP's [default socket timeout](#) is 60 seconds. This seems an unnecessarily long time to wait before deciding a request is not going to be completed.

As these are POST requests they should complete quickly - there's no data to be retrieved and returned as with a GET request.

We should be able to assume, *most of the time*, that the failure to issue a response to a request within X seconds means the host is unlikely to issue a response within a reasonable time for values of X *significantly* less than 60.

Surely hosts rarely take more than 60 seconds to respond to a simple POST request. Do they even rarely take more than 10 seconds? 5 seconds?

What might be sensible values for X in practice? Justifications accompanying suggestions would be extremely beneficial.

http

post

timeout

Share

Improve this question

Follow

edited May 23, 2017 at 10:27



Community Bot

1 • 1

asked Oct 4, 2008 at 22:12



Jon Cram

17.3k • 24 • 79 • 108

If you're uploading a file, especially from a mobile device, it could take more than 60 seconds. – [Oscar](#) Jul 4, 2011 at 17:51

2 Answers

Sorted by:

Highest score (default)





6

I would recommend setting up a test, as there are too many factors involved to give a value that will always be sensible.



A POST request sends data to be processed. How long with the processing take? This will be application/data specific.



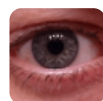
Where is the host? The user is supplying the URL, so that will be unknown. We can't know what traffic is like between your application and the host. We can't know the server load of the host.

Essentially, there is no universal sensible timeout. You have to use your own best judgment based on your specific needs. Set up a test and use that to determine your limits.

Share Improve this answer

answered Oct 4, 2008 at 23:31

Follow



Phoenix

581 ● 4 ● 15



3

Most libraries have a connect timeout and a read timeout. That is, the timeout between trying to connect to the remote server, and the timeout after sending the request, that they should wait for a response.



If this is a local web service, I would set the connect timeout low, 1 second, or less if your library supports it. If the remote service you are connecting to is unavailable



IMHO its better to return a response to the user immediately, than to allow all your worker threads to block on that remote service, causing other upstream errors.

As for the read timeout, that is trickier, you need it to be low, so you don't exhaust your pool of workers who are waiting for the remote service to return, but you also don't want it so low that it closes the connection before reading a response. That is something you'll have to test, then track as a metric when your system is in production.

[Share](#) [Improve this answer](#)

answered Oct 5, 2008 at 3:40

[Follow](#)



[Dave Cheney](#)

5,755 ● 2 ● 21 ● 24
