Subject: Re: HttpClient Execute vs ExecuteRedirect
Posted by rylek on Tue, 05 Oct 2010 15:37:34 GMT

View Forum Message <> Reply to Message

Hi there!

The main purpose of ExecuteRedirect is to jump over redirection responses. A HTTP server can respond with a result code in the range 300 .. 399 to indicate that the requested resource is available at a different URI. ExecuteRedirect detects these cases and follows the redirection chain to (hopefully) obtain the 'true' final resource. The maximum number of times to hop these redirections is the first parameter in call to ExecuteRedirect and by default is set to the symbolic constant HttpClient::DEFAULT_MAX_REDIRECT ::= 5. Each hop is a separate request and as such it should timeout whenever the server becomes inactive for HttpClient::DEFAULT_TIMEOUT_MSECS ::= 120000 msecs (the default is 2 minutes, see also HttpClient::TimeoutMsecs()).

Note that the timeout check is not entirely reliable: if the server was able to produce, say, 1 byte a minute, the HttpClient would never proclaim it inactive even if dowloading a single web page took three days. On the other hand, you can use the 'progress' parameter in calls to Execute, ExecuteRedirect or HttpClientGet, to supply a custom callback to detect additional reasons for aborting a pending web request (e.g. maximum total duration) and to tell the Http client to stop communicating with the server and return immediately with the appropriate error code.

Regards

Tomas