NETCURLv6.1

Code

Currently, and still in a one-man-army mode, the code for next verson (6.1) could be found hhereere. There's a big difference between 6.0 and 6.1 - it's more standardized and PSR-4 complient. However, the goal is to keep compatibility to 6.0 so that developers can upgrade to proper code without ripping their hair off their heads. 6.1 will be merged into the master branch as soon as it has been properly tested which is primarily done with Bamboo, Pipelines and pipelines with regular ecommerce systems.

Live information

There's a Mailinglist put up for everything concerning netcurl. That's also where you can find release information (for now). You can subscribe to the list here .

Supported Versions

Branch	Development started	Initial Release	Active Support Until	Maintenance	PHP Support
5.0	**2016-12-16**	Never	2017-08-17	2017-08-17	5.4 - 7.4 *Follows 6.0 branch.*
6.0	**2017-08-17**	**2017-09-01** Commit 397c036fc61	2020-04-01	2020-11-25	*(5.3)* 5.4 - 7.4 NOT PHP 8.0
6.1	**2019-09-01** According to commit b4dea50fd24	**2020-05-24**	-	-	5.6 - 8.0 (From >6.1.1)

Inline documentation

Autogenerated docs can be found, daily, at https://gitreport.tornevall.net/tornelib-php-netcurl-6.1/.

Compatibility

To keep compatibility with v6.0 the plan is to keep the primary class MODULE_CURL callable from a root position. It will probably be recommended to switch over to a PSR friendly structure from there, but the base will remain in 6.1 and the best way to instantiate the module in future is to call for the same wrapper as the main MODULE_CURL will use - NetWrapper (TorneLIB\Module\Network\NetWrapper) as it is planned to be the primary driver handler.

Project status

v6.1.1

Key	Summary	T	Updated	Status

No issues found

v6.1.0

Key	Summary	T	Updated	Status
NETCURL-263	Add errorhandler for multicurl	☑	May 27, 2020	DONE
NETCURL-258	NetWrapper MultiRequest	☑	May 27, 2020	DONE
NETCURL-285	Reinstate Environment but in ConfigWrapper to make wsdl transfers go non-cache vs cache, etc	☑	May 20, 2020	DONE
NETCURL-295	MultiNetwrapper (+Soap)	☑	May 12, 2020	DONE

NETCURL-278	setChain in 6.1 should throw errors when requested true	☑	May 12, 2020	DONE
NETCURL-293	Try fix proper rss parsing without garbage	☑	May 09, 2020	DONE
NETCURL-280	proxy support for curlwrapper and wrappers that is not stream wrappers	☑	May 09, 2020	DONE
NETCURL-294	Make it possible to initialize an empty curlwrapper (without url)	☑	May 09, 2020	DONE
NETCURL-292	Support basic rss+xml via GenericParser	☑	May 09, 2020	DONE
NETCURL-261	Make sure setAuthentication is a required standard in the wrapper interface	☑	May 09, 2020	DONE
NETCURL-260	Current curl implementation is only using GET and no advantage of Config	☑	May 09, 2020	DONE
NETCURL-288	Output support for XML in simpler wrappers	☑	May 09, 2020	DONE
NETCURL-291	SoapClient must be reinitialized each time it is called	☑	Apr 27, 2020	DONE
NETCURL-231	Move out MODULE_NETWORK to own repo	☑	Apr 27, 2020	DONE
NETCURL-287	Initialize simplified streamSupport	☑	Apr 25, 2020	DONE
NETCURL-283	Use setSignature (?) to make requesting clients set internal clientname/version as userAgent automatically instead of Mozilla	☑	Apr 25, 2020	DONE
NETCURL-277	Netwrapper Compatibility Service	☑	Apr 25, 2020	DONE
NETCURL-289	Open for third party identification rather than standard browser agent	☑	Apr 25, 2020	DONE
NETCURL-286	Move driver handler into own class	☑	Apr 25, 2020	DONE
NETCURL-279	Make sure setOption is useful in NetWrapper and MODULE_CURL or work has been useless	☑	Apr 22, 2020	DONE
NETCURL-234	Support immediate inclusions of network libraries	☑	Apr 22, 2020	DONE
NETCURL-271	Synchronize with netcurl 6.0 test suites	☑	Apr 22, 2020	DONE
NETCURL-247	The way we set user agent in the SSL module must be able to set in parents	☑	Apr 22, 2020	DONE
NETCURL-209	Support stream, when curl is not an option	☑	Apr 20, 2020	DONE
NETCURL-238	Make setTimeout support ms (curlopt_timeout_ms)	☑	Apr 20, 2020	DONE
NETCURL-259	High focus on curl (rebuild from 6.0)	☑	Apr 20, 2020	DONE
NETCURL-268	Add Timeouts	☑	Apr 20, 2020	DONE
NETCURL-245	Reimport soapclient	☑	Apr 20, 2020	DONE
NETCURL-244	Reimport curl module	☑	Apr 20, 2020	DONE
NETCURL-243	Reimport SSL helper	☑	Apr 20, 2020	DONE
NETCURL-276	Use a natural soapcall with call_user_func_array	☑	Apr 20, 2020	DONE

NETCURL-242	Disengage from constructor usage		Apr 20, 2020	DONE
NETCURL-264	Avoid static constants inside core functions		Apr 20, 2020	DONE
NETCURL-273	Support driverless environment		Apr 20, 2020	DONE
NETCURL-200	Confirm (by driver) that a driver is really available (update interface with requirements)		Apr 19, 2020	DONE
NETCURL-227	Migrate: getHttpHost()		Apr 19, 2020	DONE
NETCURL-265	getCurlException($curlHandle, $httpCode) - httpCode is unused. Throw on >400		Apr 17, 2020	DONE
NETCURL-249	setAuth for curl		Apr 17, 2020	DONE
NETCURL-251	setAuth for soap		Apr 17, 2020	DONE
NETCURL-269	Proxy support for stream_context		Apr 10, 2020	DONE
NETCURL-267	On http head errors (>400) and non empty bodies		Apr 08, 2020	DONE
NETCURL-255	Add static list of browsers for user-agent		Mar 06, 2020	DONE
NETCURL-232	Make exceptions global		Feb 18, 2020	DONE
NETCURL-225	NetCURL 6.1		Oct 25, 2020	DONE
NETCURL-246	Pipeline errors for PHP 7.3-7.4		Apr 20, 2020	DONE
NETCURL-274	Cached wsdl requests and unauthorized exceptions		Apr 20, 2020	DONE
NETCURL-272	getSoapEmbeddedRequest() - PHP 5.6+PHP 7.0		Apr 20, 2020	DONE
NETCURL-226	PSR4 NetCURL+Network (Phase 1)		Apr 19, 2020	DONE
NETCURL-230	Wordpress driver in prior netcurl is lacking authentication mechanisms		Apr 03, 2020	DONE

49 issues

Backstory

History

NETCURL is a network library with a backstory in the simplicity to find and extract lists of ip-addresses from websites, utilize them and register them as proxies of different types. In short, NETCURL was once built for the DNSBL API.

Present time

NetCURL has changed purpose over time. What was planned to be scraping tool became, thanks to the self selective engine that was added, a tool to automatically fetch and parse data and transform it to standard output arrays/objects. The implementation supported common communications including SOAP and partially PHP built in streams though extra drivers - and it was autoselective. If the primary driver wasn't present, it jumped to next one and so on.

The intentions with NETCURL was probably never to rebuild the wheel. Many solutions already did this, but backfired somewhere when it came to setting the utilities up. Very often developers had to make all configurations manually (Both Guzzle and Zend does this), while NETCURL was firing up a high-verbose-level-fetching communications driver automatically. In our case, we wanted to utilize as much packages as possible and choose the best preferred method without asking question. The goal? Standard output, ready to be fetched by the endpoint developer.