README.md / Installation

Installation

Recommended solution: Composer.

Alternatives: git clone this repo.

Contact and 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
.

Documents

- Version 6.1
- Exceptions handling for v6.0

NETCURL IS AND IS NOTS

- Written for 6.0

HOWTOs

Getting started

- MODULE_CURL 6.1
- Getting started: Individual Modules

Obsolete documents

- MODULE_CURL 6.0

XML, CURL, SOAP, JSON

In apt-based systems, extra libraries can be installed with commands such as:

```
apt-get install php-curl php-xml php-json php-soap
```

The module installation itself

This is the recommended way (and only officially supported) of installing the package.

- Get composer.
- Run composer as shown below.

```
composer require tornevall/tornelib-php-netcurl
```

If you wish, you can alternative also do one of those requirements, when the tags is ready:

```
composer require tornevall/tornelib-php-netcurl:^6.1
# or
composer require tornevall/tornelib-php-network:^6.1
```

Installing the network module instead will make features such as getGitTagsByUrl fully available.

Module Information

Compatibility

This library is built to work with PHP 5.6 and higher (I prefer to follow the live updates of PHP with their EOL's - check it here). The Bamboo-server has a history which makes PHP from 5.4 available on demand. However, autotests tend to fail on older PHP's so as of march 2020 all tests lower than 5.6 is disabled.

However, it is not that easy. The compatibility span **has** to be lower as the world I'm living in tend to be slow. If this module is built after the bleeding edge-principles, that also means that something will blow up somewhere. It's disussable whether that's something to ignore or not, but I think it's important to be supportive regardless of end of life-restrictions (but not too far). When support ends from software developers point of view, I see a perfect moment to follow that stream. This is very important as 2019 and 2020 seems to be two such years when most of the society is forcing movement forward.

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.

Requirements and dependencies

In its initial state, there are basically no requirements as this module tries to pick the best available driver in runtime.

Library Support

Current

- curl
- The simplest form of streamdriver and the binary safe file_get_contents (instead of the fopen-drivers that is based on the same system).
- SoapClient

Pending/in progress

- RSS feeds

Removed

- Guzzle
- Zend
- Sockets
- Wordpress (At least for now)

The reason of the above removals is simple: Both guzzle, zend and wordpress drivers are actually based on the same methods that is already implemented in this package in the form of curl and stream-support. Wordpress for example, first tries to utilize curl support is available and then fails over to the stream support. This package is using file_get_contents after an important decision to be binary safe, as this method is. The fopen/fread-methods are not necessarily binary safe so in the first release, this support has been skipped. As for the other drivers, zend and guzzle, there are similar ways of running curl=>stream.

How about sockets?

This driver requires more, so this is put on hold.

Dependencies, not required, but recommended

- SSL: OpenSSL or similar (if doing https requests).
- SOAP: SoapClient and XML-drivers (if doing https requests).
- CURL (if you prefer curl before streams).
- allow_url_fopen (if you have no access to curl, this has to be enabled).

Changes

Version 6.1 follows the standard of what's written in 6.0 - there is a primary module that handles all actions regardless of which driver installed. However, 6.0 is built on a curl-only-core with only halfway fixed PSR-supported code. For example, autoloading has a few workarounds to properly work with platforms that requires PSR-based code. In the prior release the class_exists-control rather generates a lot of warning rather than "doing it right". In 6.1 the order of behaviour is changed and all curl-only-based code has been restructured.

Breaking changes?

No. Version 6.1 is written to reach highest compatibility with v6.0 as possible, but with modernized code and PSR-4. Make sure you check out https://docs.tornevall.net/x/DoBPAw before deciding to run anything as older PHP-releases could be incompatible. However, if they are, so are probably you.