

PHP Curl Class makes it easy to send HTTP requests and integrate with web APIs <https://www.phpcurlclass.com/>

#php #curl #class #php-curl #api #restful #json #web-services #client #framework #http-client #http #requests #rest #web-scraper #web-scraping #xml

1,031 commits

1 branch

32 releases

36 contributors

Unlicense

Branch: master

















New pull request

Create new file

Upload files

Find file

Clone or download

 zachborboa	Update @return documentation	Latest commit 6efebcd 3 days ago
 .github	Add contributing and issue templates	2 years ago
 docs	Remove outdated docs; See examples instead	a year ago
 examples	Fix #305: Implement Curl::getResponseCookies()	3 months ago
 scripts	Rename callbacks	16 days ago
 src/Curl	Update @return documentation	3 days ago
 tests	Skip post-redirect-get test on hhvm-nightly	15 days ago
 www	Add screencast	a year ago
 .gitattributes	Add trailing slashes to directories	4 months ago
 .gitignore	Add defaultDecoder for parseResponse	2 years ago
 .travis.yml	Use Trusty for all builds except PHP 5.3	7 months ago
 LICENSE	Initial commit	5 years ago
 README.md	Fix #305: Implement Curl::getResponseCookies()	3 months ago
 SECURITY.md	Clean up	a year ago
 TROUBLESHOOTING.md	Use full example for enabling verbose mode	4 months ago
 composer.json	Improve description	a year ago

README.md

PHP Curl Class: HTTP requests made easy

release **v8.0.1** license **Unlicense** build **failing** downloads **749k**

PHP Curl Class makes it easy to send HTTP requests and integrate with web APIs.

```
[ "url" ] =>
string(23) "https://httpbin.org/get"
}
php >
php >
php > echo 'now for concurrent requests';
now for concurrent requests
php >
php > use \Curl\MultiCurl;
php >
php > $multi_curl = new \Curl\MultiCurl();
php > $multi_curl->success(function($instance) {
php ( echo 'call to ' . $instance->url . ' was successful' . "\n";
php ( });
php >
php > echo 'queue requests';
queue requests
php >
php > $multi_curl->addGet('https://httpbin.org/get');
php > $multi_curl->addPost('https://httpbin.org/post');
php > $multi_curl-
```

- [Installation](#)
 - [Requirements](#)
 - [Quick Start and Examples](#)
 - [Available Methods](#)
 - [Security](#)
 - [Troubleshooting](#)
 - [Run Tests](#)
 - [Contribute](#)
-

Installation

To install PHP Curl Class, simply:

```
$ composer require php-curl-class/php-curl-class
```

For latest commit version:

```
$ composer require php-curl-class/php-curl-class @dev
```

Requirements

PHP Curl Class works with PHP 5.3, 5.4, 5.5, 5.6, 7.0, 7.1, and HHVM.

Quick Start and Examples

More examples are available under [/examples](#).

```
require __DIR__ . '/vendor/autoload.php';

use \Curl\Curl;

$curl = new Curl();
$curl->get('https://www.example.com/');

if ($curl->error) {
    echo 'Error: ' . $curl->errorCode . ': ' . $curl->errorMessage . "\n";
} else {
    echo 'Response: ' . "\n";
    var_dump($curl->response);
}
```

```
// https://www.example.com/search?q=keyword
$curl = new Curl();
$curl->get('https://www.example.com/search', array(
    'q' => 'keyword',
));
```

```
$curl = new Curl();
$curl->post('https://www.example.com/login/', array(
    'username' => 'myusername',
    'password' => 'mypassword',
));
```

```
$curl = new Curl();
$curl->setBasicAuthentication('username', 'password');
$curl->setUserAgent('MyUserAgent/0.0.1 (+https://www.example.com/bot.html)');
$curl->setReferrer('https://www.example.com?url=https%3A%2F%2Fwww.example.com%2F');
$curl->setHeader('X-Requested-With', 'XMLHttpRequest');
$curl->setCookie('key', 'value');
$curl->get('https://www.example.com/');

if ($curl->error) {
```

```

        echo 'Error: ' . $curl->errorCode . ': ' . $curl->errorMessage . "\n";
    } else {
        echo 'Response: ' . "\n";
        var_dump($curl->response);
    }

    var_dump($curl->requestHeaders);
    var_dump($curl->responseHeaders);

```

```

$curl = new Curl();
$curl->setOpt(CURLOPT_FOLLOWLOCATION, true);
$curl->get('https://shortn.example.com/bHbVsP');

```

```

$curl = new Curl();
$curl->put('https://api.example.com/user/', array(
    'first_name' => 'Zach',
    'last_name' => 'Borboa',
));

```

```

$curl = new Curl();
$curl->patch('https://api.example.com/profile/', array(
    'image' => '@path/to/file.jpg',
));

```

```

$curl = new Curl();
$curl->patch('https://api.example.com/profile/', array(
    'image' => new CURLFile('path/to/file.jpg'),
));

```

```

$curl = new Curl();
$curl->delete('https://api.example.com/user/', array(
    'id' => '1234',
));

```

```

// Enable all supported encoding types and download a file.
$curl = new Curl();
$curl->setOpt(CURLOPT_ENCODING, '');
$curl->download('https://www.example.com/file.bin', '/tmp/myfile.bin');

```

```

// Case-insensitive access to headers.
$curl = new Curl();
$curl->download('https://www.example.com/image.png', '/tmp/myimage.png');
echo $curl->responseHeaders['Content-Type'] . "\n"; // image/png
echo $curl->responseHeaders['CoNtEnt-TyPe'] . "\n"; // image/png

```

```

// Clean up.
$curl->close();

```

```

// Example access to curl object.
curl_setopt($curl->curl, CURLOPT_USERAGENT, 'Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1)');
curl_close($curl->curl);

```

```

require __DIR__ . '/vendor/autoload.php';

use \Curl\MultiCurl;

// Requests in parallel with callback functions.
$multi_curl = new MultiCurl();

$multi_curl->success(function($instance) {
    echo 'call to "' . $instance->url . '" was successful.' . "\n";
    echo 'response: ' . "\n";
    var_dump($instance->response);

```

```

});
$multi_curl->error(function($instance) {
    echo 'call to "' . $instance->url . '" was unsuccessful.' . "\n";
    echo 'error code: ' . $instance->errorCode . "\n";
    echo 'error message: ' . $instance->errorMessage . "\n";
});
$multi_curl->complete(function($instance) {
    echo 'call completed' . "\n";
});

$multi_curl->addGet('https://www.google.com/search', array(
    'q' => 'hello world',
));
$multi_curl->addGet('https://duckduckgo.com/', array(
    'q' => 'hello world',
));
$multi_curl->addGet('https://www.bing.com/search', array(
    'q' => 'hello world',
));

$multi_curl->start(); // Blocks until all items in the queue have been processed.

```

More examples are available under [/examples](#).

Available Methods

```

Curl::__construct($base_url = null)
Curl::__destruct()
Curl::__get($name)
Curl::attemptRetry()
Curl::beforeSend($callback)
Curl::buildPostData($data)
Curl::call()
Curl::close()
Curl::complete($callback)
Curl::delete($url, $query_parameters = array(), $data = array())
Curl::download($url, $mixed_filename)
Curl::error($callback)
Curl::exec($ch = null)
Curl::execDone()
Curl::get($url, $data = array())
Curl::getCookie($key)
Curl::getInfo($opt = null)
Curl::getOpt($option)
Curl::getResponseCookie($key)
Curl::getResponseCookies()
Curl::head($url, $data = array())
Curl::options($url, $data = array())
Curl::patch($url, $data = array())
Curl::post($url, $data = array(), $follow_303_with_post = false)
Curl::progress($callback)
Curl::put($url, $data = array())
Curl::removeHeader($key)
Curl::search($url, $data = array())
Curl::setBasicAuthentication($username, $password = '')
Curl::setConnectTimeout($seconds)
Curl::setCookie($key, $value)
Curl::setCookieFile($cookie_file)
Curl::setCookieJar($cookie_jar)
Curl::setCookieString($string)
Curl::setCookies($cookies)
Curl::setDefaultDecoder($mixed = 'json')
Curl::setDefaultJsonDecoder()
Curl::setDefaultTimeout()
Curl::setDefaultUserAgent()
Curl::setDefaultXmlDecoder()
Curl::setDigestAuthentication($username, $password = '')
Curl::setHeader($key, $value)
Curl::setHeaders($headers)
Curl::setJsonDecoder($mixed)
Curl::setMaxFileSize($bytes)
Curl::setOpt($option, $value)
Curl::setOpts($options)
Curl::setPort($port)
Curl::setReferer($referer)
Curl::setReferrer($referrer)

```

```

Curl::setRetry($mixed)
Curl::setTimeout($seconds)
Curl::setUrl($url, $mixed_data = '')
Curl::setUserAgent($user_agent)
Curl::setXmlDecoder($mixed)
Curl::success($callback)
Curl::unsetHeader($key)
Curl::verbose($on = true, $output = STDERR)
MultiCurl::__construct($base_url = null)
MultiCurl::__destruct()
MultiCurl::addCurl(Curl $curl)
MultiCurl::addDelete($url, $query_parameters = array(), $data = array())
MultiCurl::addDownload($url, $mixed_filename)
MultiCurl::addGet($url, $data = array())
MultiCurl::addHead($url, $data = array())
MultiCurl::addOptions($url, $data = array())
MultiCurl::addPatch($url, $data = array())
MultiCurl::addPost($url, $data = array(), $follow_303_with_post = false)
MultiCurl::addPut($url, $data = array())
MultiCurl::addSearch($url, $data = array())
MultiCurl::beforeSend($callback)
MultiCurl::close()
MultiCurl::complete($callback)
MultiCurl::error($callback)
MultiCurl::getOpt($option)
MultiCurl::removeHeader($key)
MultiCurl::setBasicAuthentication($username, $password = '')
MultiCurl::setConcurrency($concurrency)
MultiCurl::setConnectTimeout($seconds)
MultiCurl::setCookie($key, $value)
MultiCurl::setCookieFile($cookie_file)
MultiCurl::setCookieJar($cookie_jar)
MultiCurl::setCookieString($string)
MultiCurl::setCookies($cookies)
MultiCurl::setDigestAuthentication($username, $password = '')
MultiCurl::setHeader($key, $value)
MultiCurl::setHeaders($headers)
MultiCurl::setJsonDecoder($mixed)
MultiCurl::setOpt($option, $value)
MultiCurl::setOpts($options)
MultiCurl::setPort($port)
MultiCurl::setReferer($referer)
MultiCurl::setReferrer($referrer)
MultiCurl::setRetry($mixed)
MultiCurl::setTimeout($seconds)
MultiCurl::setUrl($url)
MultiCurl::setUserAgent($user_agent)
MultiCurl::setXmlDecoder($mixed)
MultiCurl::start()
MultiCurl::success($callback)
MultiCurl::unsetHeader($key)
MultiCurl::verbose($on = true, $output = STDERR)

```

Security

See [SECURITY](#) for security considerations.

Troubleshooting

See [TROUBLESHOOTING](#) for troubleshooting.

Run Tests

To run tests:

```

$ git clone https://github.com/php-curl-class/php-curl-class.git
$ cd php-curl-class/
$ composer update
$ ./tests/run.sh

```

Contribute

1. Check for open issues or open a new issue to start a discussion around a bug or feature.

2. Fork the repository on GitHub to start making your changes.
3. Write one or more tests for the new feature or that expose the bug.
4. Make code changes to implement the feature or fix the bug.
5. Send a pull request to get your changes merged and published.