

Features

- API calls
- No Dependencies
- [Super Fast](#)
- Extensible
- [Tested](#) in PHP 5.3 to 7.4
- [Highly secured with 512Mb encryption key](#)

Making an API call using PHP

If you are going to make a call using PHP then recommended secure curl call settings are as follows:

First install the curl dependency [using composer]:

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

Or download the [latest release](#) and include `Curl.php`

Environment Requirements

PHP Curl Class running on a PHP 5.3, 5.4, 5.5, 5.6, 7.0, 7.1, 7.2, 7.3, 7.4, and HHVM.

API Call requirements

Variables containing your encryption key, username and password

```
$_API_KEY = "AbcXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX";// This encryption key is provided by ProLasku
$_API_USERNAME = "your_api_user_name";
$_API_PASSWORD = "your_api_password";
```

Example:

Steps:

- Setting an "Authorization1" header
- Initiating a call to get data from the remote server
- Receiving the response from the remote server and error reporting
- Parsing the response into php array

Setting an "Authorization1" header

```
// initiating the curl class
$curl = new Curl();
// setting authorization header and encrypting it with a basic php function "base64_encode"
$curl->setHeader('Authorization1', base64_encode($_API_KEY));
```

Initiating a call to get data from the remote server:

```
$curl->post('https://easycms.pro/public_api/'.$_API_CALL.'/'.$_API_CALL, array(
    'username' => $_API_USERNAME,
    'password' => $_API_PASSWORD,
));
```

Receiving the response from the remote server and error reporting

```
// dealing with error if there is any
$error = $response = false;
if($curl->error):
    $error = $curl->errorCode . ': ' . $curl->errorMessage . "\n";
else: // otherwise just proceed to the response
    $response = $curl->response;
endif;
```

Parsing the response into php array

```
echo '<pre>';
echo '===== ' . "\n";
if($error):
    print_r($error);
else: // otherwise just proceed to the response
    print_r(json_decode($curl->response, true));
endif;
```

Security

Prolasku is capable of escaping user-input within the HTML that it generates. Additionally PL will apply sanitisation to additional scripting vectors (such as scripting link destinations) that are introduced by the request and also the response syntax itself.