



GENERAL USAGE OF CDPAY API

All methods must be POST type except the method “eshop_payment_v2” and must be setup as follows:

Http header for POST method

API-Key = API key

API-Sign = Message signature using HMAC-SHA512 of (URI path + '?method=<method>' + SHA256(nonce + timestamp + POST data)) and secret key.

Content type of the request must be set to: **Content-type: application/json**

POST data required and send in json format:

nonce = always increasing unsigned 64 bit integer

timestamp = current Unix timestamp.

Note that there is no way to reset the nonce to lower value. The only way to reset the nonce is to reset your secret key in CDPAY portal site which will clear your nonce to 0.

The time on your server need to be synced to correctly work with the API.

All responses will contain two JSON variables:

- **error** - integer value 0 or 1. 0=no error, 1=error
- **error_msg** - string value - description of error.

Note: for test environment please use the URL:

<https://www.cryptodiggerstest.eu/api/api2.php>



API methods

NEW PAYMENT

http://www.cdpay.eu/api2.php?method=new_address

JSON params:

timeout - integer value, $5 \leq \text{timeout} \leq 30$

order_id - string value, length 1-50, only alphanumeric characters + '-', '_'

amount - decimal value, $0.10 \leq \text{amount} \leq 20000$

currency - integer value:

- EUR=1
- USD=2
- CZK=16
- GBP=3
- CAD=4
- AUD=5
- JPY=9
- CNY=13
- AED=17
- PLN=18
- CHF=23

currency_crypto - integer value:

- BTC=6
- LTC=8
- DASH=19
- XMR=20
- ZEC=21
- BCASH=24

JSON request example:

```
{ "timeout":10, "order_id":"Test123456789asfda", "amount":10, "currency":1, "currency_crypto":8, "wait":0, "timestamp":1531131753, "nonce":"1531131753238288" }
```

JSON response example:

```
{ "address": [ { "address_value_out":"n4pKTuQ9iAMe5sb4QAm1kiuCLZfpqY3bPE", "amount_out":"0.14491204", "iframe_id":"d1a5bfe7-8362-11e8-a8c0-00155d261518", "currency_id_out":8, "currency_out":"LTC", "Msg":"OK" } ], "error":0, "error_msg":"" }
```



IFRAME ESHOP

https://www.cdpay.eu/api2.php?method=eshop_payment_v2&iframe=941b192b-88eb-11e8-b8eb-00155d261518&amount=10

This iframe should be show as an iframe on your page. It contain the HTML code of the in-page payment frame.

GET params:

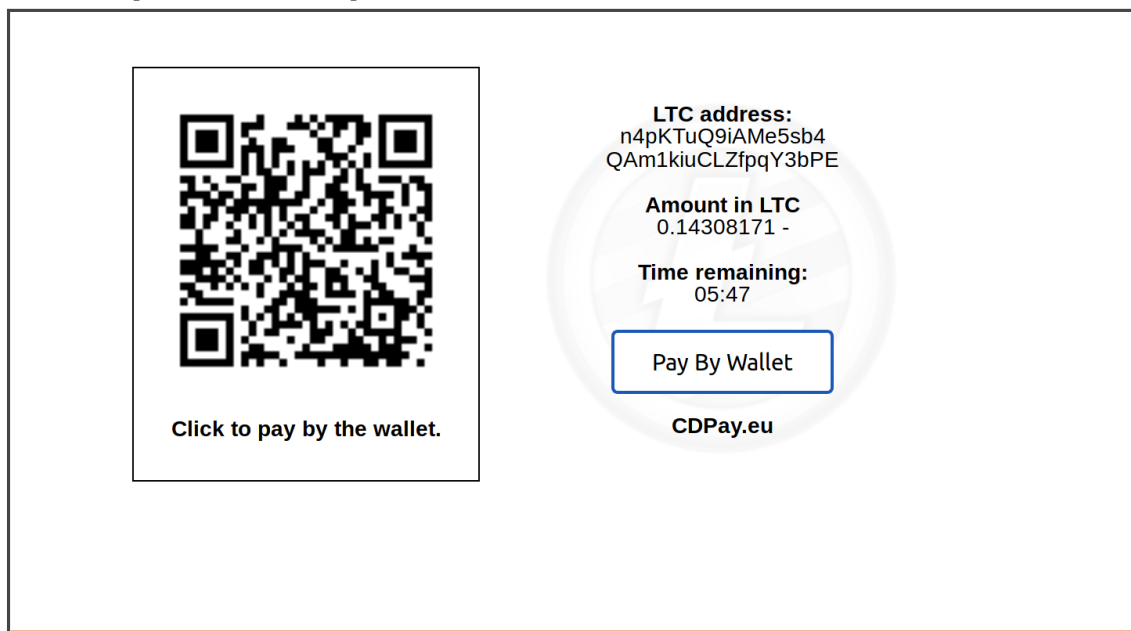
iframe - unique id from New Payment method.

amount - decimal value, 0.10 >=amount<=20000

(Website tag)

```
<iframe src="https://www.cdpay.eu/api2.php?method=eshop_payment_v2&iframe=941b192b-88eb-11e8-b8eb-00155d261518&amount=10.01" width="720px" height="400px"></iframe>
```

Iframe response example:



GET STATUS OF PAYMENT

http://www.cdpay.eu/api2.php?method=get_iframe_status

JSON params:

iframe - unique id from New Payment method.



JSON request example:

```
{"iframe":"05490a8e-8364-11e8-a8c0-00155d261518","timestamp":1531132624,"nonce":"1531132624784835"}
```

JSON response example:

```
{"error":0,"error_msg":"","status_msg":"Payment not received","status_id":4}
```

Status_id and **status_msg** are localized for EN, DE, SK, CZ languages.

Status codes:

- 0** - Payment method not available at this time
- 1** - Payment was successfully received
- 2** - You have sent incorrect amount of virtual coins to the generated address. Please contact [eshop](#) for more information
- 3** - Waiting for payment
- 4** - Payment not received
- 5** - Order ID does not exist
- 6** - Payment method not available at this time. Internal Error
- 98** - Access denied

GET CURRENT EXCHANGE RATE:

http://www.cdpay.eu/api2.php?method=get_exch_rate

JSON params:

currency - integer value:

- EUR=1
- USD=2
- CZK=16
- GBP=3
- CAD=4
- AUD=5
- JPY=9
- CNY=13
- AED=17
- PLN=18
- CHF=23

currency_crypto - integer value:

- BTC=6
- LTC=8
- DASH=19
- ZEC=21



- BCASH=24

JSON request example:

```
{"currency":1,"currency_crypto":8,"timestamp":1531133844,"nonce":"1531133844367337"}
```

JSON response example:

```
{"exch_rate":"69.4600000","error":0,"error_msg":""}
```

CALLBACK

Callback functionality is used to update the status of the payment. The standard processing of the payment is following:

- 0** - payment not received
- 1** - payment received but not confirmed
- 2** - payment received and confirmed
- 3** - payment received but with incorrect amount

The callback is executed from the CDPay API server every few seconds to update the status of the payment. The HTTP header contain the signature to verify whether the data was not changed during transmission.

The process of verification is similar to the standard API calls.

API-Sign = Message signature using HMAC-SHA512 of (Callback URI path + SHA256(nonce +timestamp+ POST data)) and secret key.

Example in PHP:

```
$sign = base64_encode(hash_hmac('sha512',  
$eshop_url_callback.hash('sha256', $data['nonce'] . $data['timestamp'] .  
http_build_query($data, '', '&'), true), $this->secret, true));
```

The data sent to the callback at the eshop side must be implemented as follows:

Content type for the callback function: **“application/json”**.

txid - string value, contain the transaction id of the virtual transaction

order_id - string value, length 1-50, only alphanumeric characters + '-', '_'

amount - decimal value, 0.10 >=amount<=20000

currency - integer value:

- EUR=1
- USD=2
- CZK=16
- GBP=3
- CAD=4



- AUD=5
- JPY=9
- CNY=13
- AED=17
- PLN=18
- CHF=23

currency_crypto - integer value:

- BTC=6
- LTC=8
- DASH=19
- XMR=20
- ZEC=21
- BCASH=24

status - integer value, 0-3

timestamp - current time stamp from CDPay API

nonce - integer value, always incremented to previous one.

JSON example:

```
{"txid":"0aa65e9cac6425f640c6c4b76398325d2c6cc44f6b66d8bf752a46de914eac43","amount":10,"order_id":"Order1","status":1,"currency":1,"currency_crypto":8}
```

Call back page must answer with HTTP code 200, with only a value true/false.

Disclaimer: any bad implementation, not using SSL connection and additional security on your e-commerce platform could lead to exposing your secrets to an attacker and thus your digital currencies could be threatened by stealing with no warranty on CryptoDiggers s.r.o. side !