USAGE OF CDPAY API

with redirect page

This document describes usage of CryptoDiggers CDPAY API with redirect page (not using iframe option) to implement supported cryptocurrency payment to your e-commerce site.

CDPAY environments:
Test: https://www.cryptodiggerstest.eu/api
Live: https://www.cdpay.eu

Redirect endpoint: redirect.php

Payment through redirect page is done in 3 steps.
1. Generate redirect link to show payment gateway
2. Create a return page for success/failure/timeout of the payment
3. Create a callback page for success/failure/timeout of the payment

Redirect page HTTP method must be GET. All parameters are mandatory:

timeout – integer value, 5 - 30

order_id – string value, length 1-50, allowed alphanumeric characters 0-9, +, '-', '_'

amount – decimal value, 0.10 - 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
nonce	-	always increasing unsigned 64 bit integer
timestamp	-	current Unix timestamp.
order_id	-	string value, length 1-50, allowed alphanumeric characters 0-9, +, '-', '_'
returnAddress	-	User defined redirect page for version V2, must be URL encoded
api-key	-	CDPAY user API key
api-sign	-	Message signature base64 encoded using HMAC-SHA512 consisting of:*Test/Live CDPAY URL +'?' + < all get parameters except api- signature >*

Example for redirect version 1 (cryptocurrency defined in advance):

https://www.cdpay.eu/redirect.php?order_id=asidf111as&amount=10.1&currency=1&currency_crypto=6&wait=0&timeout=6&api-key=c70ea0e0-1454-11e6-8eeb-00155d00c800&timestamp=1534498676&nonce=1534498676199779&api-sign=ejZ38gAtuZh2AK%2Bdch1GQhBfFiBuXEKToeB5%2FJHOzGLUxojrHyNcRDsSSEmMfhb6OOyN4UJRiDqQYFAMExFcNg %3D%3D

Example for redirect version 2 (cryptocurrency defined by user):

https://www.cdpay.eu/redirect2.php?order_id=1223&amount=5&currency=1&wait=0&timeout=15&api-key=bc5228b4-865e-11eb-8cfc-00155dfb8403&timestamp=1641650126&nonce=1641650126511172&api-sign=wS9u5T%2B16tdQ%2F2FC%2BO1efvZ%2FtQ633u5SFCOQLCkHPC3gIRoblVk29vKtQvPsjVjCu90rHwA%2Fq%2Bly4hOoATcsCw%3D%3D

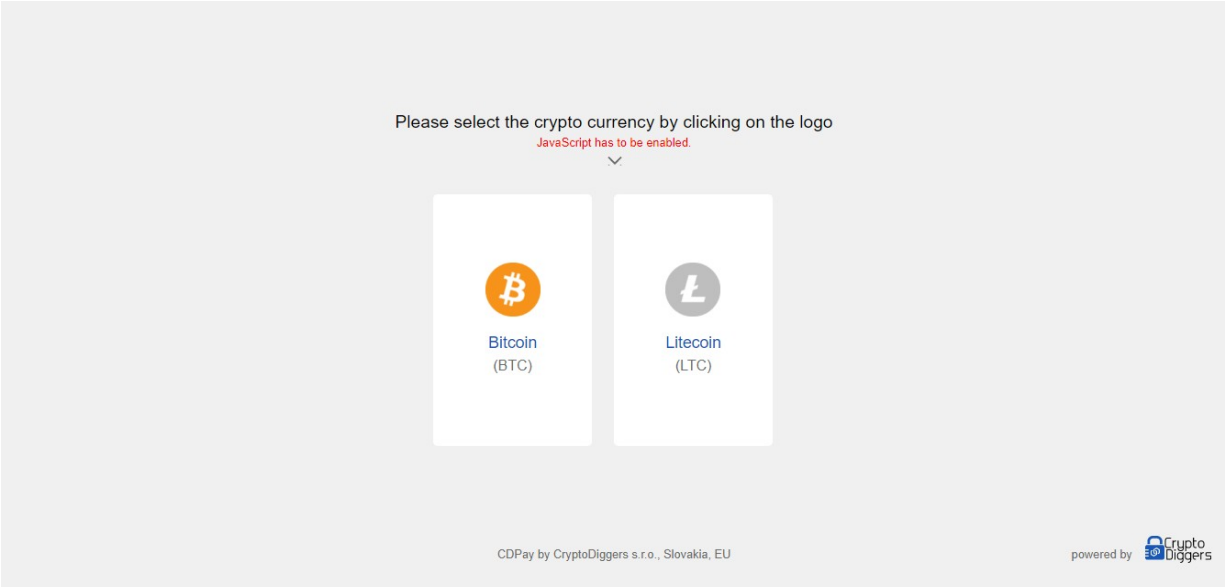
Example for redirect version 2 (cryptocurrency defined by user with redirect page definition):

https://www.cdpay.eu/redirect2.php?order_id=1223&amount=5&currency=1&wait=0&timeout=15&returnAddress=https://www.your_redirect_page.com&api-key=bc5228b4-865e-11eb-8cfc-00155dfb8403&timestamp=1641650126&nonce=1641650126511172&api-sign=wS9u5T%2B16tdQ%2F2FC%2BO1efvZ%2FtQ633u5SFCOQLCkHPC3gIRoblVk29vKtQvPsjVjCu90rHwA%2Fq%2Bly4hOoATcsCw%3D%3D

Example how to create API-Sign parameter in PHP for redirect version 1:

```
const CDPAY_TEST_WEB="https://www.cryptodiggerstest.eu/api/redirect.php";
$apikey='Vaše API key';
$seckey='Váš SecKey';
$inputs["order_id"]='OrderTest-06';
$inputs["amount"]=10.10;
$inputs["currency"]=1;
$inputs["currency_crypto"]=6;
$inputs["wait"]=0;
$inputs["timeout"]=5;
$inputs["api-key"]=$apikey;
$inputs["timestamp"] = time();
$nonce = explode(' ', microtime());
$inputs['nonce'] = $nonce[1] . str_pad(substr($nonce[0], 2, 6), 6, '0');
$sign = base64_encode(hash_hmac('sha512', CDPAY_TEST_WEB .'?'. http_build_query($inputs, '', '&'), $seckey, true));
$inputs["api-sign"]=$sign;
echo "<a href='".htmlentities(CDPAY_TEST_WEB .'?'. http_build_query($inputs, '', '&'))."'
target='_blank'>".htmlentities(CDPAY_TEST_WEB .'?'. http_build_query($inputs, '', '&'))."</a>";
```

Output for redirect V2 with user defined cryptocurrency:



Please select of supported and profile saved cryptocurrencies.

Send coins to process the showed payment.

2. Picture above showing the return page (in redirect version V2 could be defined in the API call using returnAddress paramter) during payment processing. A return URL link is called in case the payment was successful / unsuccessful or payment was not received. The processing of the return page has a HTTP method GET.

Get parameters:

nonce - always increasing unsigned 64 bit integer

timestamp - current Unix timestamp.

order_id - string value, length 1-50,
allowed alphanumeric characters 0-9, +, '-', '_'

status - payment status
- **0** – payment not received
- **1** – payment received but not confirmed
- **3** – payment received but with incorrect amount

api-sign - Message signature base64 encoded using HMAC-SHA512 consisting of:
Test/Live CDPAY URL +'?' + <all get parameters except api-signature>

Example:

https://www.eshopcallback.com/index.php?a=dsfdsf&timestamp=1534505380&nonce=1534505380020122&order_id=asidf111as&status=1&sign=BPDc43g2LjweLZ6gfkHBOU82EEQhh%2F9q%2BH4zGpoovU9PSgNqNePu3m0r3Wsi2ypeH 45I0EoCEgMBtDvoEyRn%2FA%3D%3D

Example how to create API-Sign to validate return values:

```
$request["timestamp"] = time();
$nonce = explode(' ', microtime());
$request['nonce'] = $nonce[1] . str_pad(substr($nonce[0], 2, 6), 6,
'0'); $request['order_id']='Your order ID'; $request['status']='payment
status id';
$sign = base64_encode(hash_hmac('sha512', $user_callback_url. (strpos($user_callback_url,
'?')===false?'?':'&') .http_build_query($request, '', '&'),$SecKey, true));
$request['sign']=$sign;
$url=$user_callback_url.(strpos($_SESSION["user_callback"],'?')===false?'?':'&').http_build_query($request, '', '&');
```

3. **CALLBACK**

API-Key = API key
API-Sign = Message signature using HMAC-SHA512 consisting of:

URI path +'?'+ SHA256(nonce + timestamp + POST data) and secret key.

Content type of the request must be set to: **Content-type: application/json**
nonce - always increasing unsigned 64 bit integer **timestamp** - current Unix timestamp. **POST data** - required and send in JSON format:

Note that there is no way to reset the nonce to lower value. The only way to reset the nonce is to reset your secrete 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

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 15s 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 consisting 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', $nonce . $timestamp . http_build_query($data, '', '&'), true), $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,
allowed alphanumeric characters 0-9, +, '-', '_'

amount – decimal value, 0.10 - 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":"Order 1","status":1,"currency":1,"currency_crypto":8}

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