

CDPAY digital currency payment gateway
CryptoDiggers s.r.o.

Author: CryptoDiggers s.r.o.

Version: v1.2

Last edit: 13.06.2018



CDPAY calls for the payment

Our payment gateway can be implemented as iframe to any HTML page. To do so it is required to make two API calls:

1. Background call to create iframe id. This call is done with your API key and cannot be visible on the page
2. The second call is to show the iframe on the page. In case the iframe is called multiple times it returns either:
 - a. If timeout was not reached it counts to end of the timeout
 - b. If timeout was reached it start the countdown again and recalculate the amount according the current exchange rate when it is called.
 - c. In case the payment was received the API will return a screen „Payment processed. “

How to implement callback functionality and how it works (STEP BY STEP):

1. How to setup callback

- a)** In order to use callback functionality on your e-commerce web site you need to program callback listening to below specified parameters on your custom e-commerce portal or other E-shop like the following (Magento 1.9, OpenCart 2.0, 2.2, Woocommerce) . Callback needs to update your order ID in your E-shop/E-commerce portal to state of “Paid / Processed” after reaching needed confirmation count for given digital currency. Therefore is very important to implement it correctly and securely. Otherwise you could be attacked by double spending if you update the order directly in status “Processed” before.

Statuses of order needs to be setup following: first “*Processing*” and after reaching confirmation count called by callback functionality set to “*Paid/Processed*”.

- b)** To make the callback operational you need to specify the web site URL (using custom e-commerce portal you need to place the whole URL you prepared and use type “custom”) in the CDPAY portal in your profile in the section “*User -> Add Callback*”.
- c)** Callback needs to specify so called “Callback Secret” which is pass phrase to be mixed (salted) the hash transaction ID of the “coin” (BTC, LTC, DASH, etc.) and your secret pass phrase. This hash is using SHA512 standard hashing function put in the cdp_hash parameter send by callback to your site. This needs additional action of hashing SHA512 on your side to create SHA512 hash from your passphrase



- (need to be saved on YOUR side and also at CDPAY in your profile) and `txn_id` parameter sent by callback. Afterwards those two hashed need to be compared and callback must be accepted only in case of exact match of those two hash strings. Otherwise you are at risk and attackers could do some so called man-in-the-middle attacks.
- d) In case you are our merchant we require from you to specify only the URL path which need to be called with callback functionality by CDPAY backend software.
 - e) In case you are administering your profile alone, you can specify this setting in [CryptoDiggers](#) portal. We require to setup callback URL for each digital currency you are supporting and implementing to your eshop portal.
 - f) Example of how to setup callback on Magento E-shop please follow this video on [YOUTUBE](#) [01:40].

2. Callback parameters

Callback return codes and parameters are send in **POST attributes**.

The attributes are:

- a) **cdp_hash** -> this attribute contains = *"your secret key" + "txn_id"* sent by callback, hashed with SHA512
- b) **txn_id** -> this contain the wallet TransactionID. It is highly recommended to check the sha512 value of your key and this value with the `cdp_hash` value.
- c) **amount** - > the real amount of cryptocurrency
- d) **status** -> status of processing
 - i. **1 - payment received but not confirmed**
 - ii. **2 - payment received and confirmed**
 - iii. **3 - payment received but with wrong amount**
- e) **order_id** -> your internal order id from the E-shop order.
You must ensure that this order id is always unique for every order!

STEP 1



GET NEW PAYMENT ADDRESS

Request:

<https://www.cdpay.eu/api.php?>

apikey=YOUR_API_KEY&**a=new_address**&**timeout**=15&**order_id**=Order6661&**amount**=5.00&**currency**=1&**currency_crypto**=8&**wait**=0

Response:

```
{"address":
```

```
{"address_value_out":"LS2RuWYejhSUGBE45kWRheLhpnYGBg7SRq","amount_out":  
0.03262017","iframe_id":"ede0ee2a-9b2b-11e4-b228-  
00155d006403","currency_id_out":8,"currency_out":"ltc","Msg":"OK"},"error":0,"erro  
r_msg":""}
```

- **timeout** - allowed values from 10 to 30, only integer (minutes)
- **order_id** - Customer order id from eshop - can contain only (a-z,0-9,-,_) case insensitive, up to 20 chars
- **amount** - Fiat amount, decimal separator must be “.”
- **currency** - Fiat currency allowed values (1,2,3,4,5,9,13,16,17) - EUR=1, USD=2, GBP=3, CAD= 4, AUD=5, JPY=9, CNY=13, CZK=16, AED=17, 18=PLZ, 22=VND, 23=CHF
- **currency_crypto** - Crypto currency allowed values (6,8,19,21,24) - 6=BTC, 8=LTC, 19=DASH, ZCASH=21, BCASH=24
- **wait** -wait for confirmations - allowed values (0,1) - **0=No**, 1=Yes

STEP 2

SHOW QR CODE and PAYMENT DETAILS IN IFRAME

Request:



https://www.cdpay.eu/api.php?iframe=17a2307c-0fd2-11e6-b5de-00155d000116&a=eshop_payment_v2&amount=5.00

Response:



iframe - UUID generated by function

"apikey=YOUR_API_KEY&a=eshop_payment_v2"

amount - Fiat amount, decimal separator must be "."

If there is a need to change any other parameter except the amount the user is required to call again the „1. Create iframe with payment details“ in this manual

STEP 3

CHECK PAYMENT STATUS BY IFRAME ID

Request:

http://www.cdpay.eu/api.php?iframe=17a2307c-0fd2-11e6-b5de-00155d000116&a=get_iframe_status



Response:

In case the payment was received:

```
{"error":0,"error_msg":"","status_msg":"Payment processed, please finish your order.,"status_id":1}
```

Customer sent incorrect amount to address, payment must be reviewed by

eshop: {"error":0,"error_msg":"","status_msg":" You have sent incorrect amount of virtual coins to the generated address. Please contact eshop for more information.,"status_id":2}

Payment not yet received, countdown still active:

```
{"error":0,"error_msg":"","status_msg":"Waiting for payment.,"status_id":4}
```

Payment not received to iframe id:

```
{"error":0,"error_msg":"","status_msg":"Payment not received.,"status_id":5}
```

Iframe id does not exist, please can the API function „Create iframe with payment details“:

```
{"error":0,"error_msg":"","status_msg":"Iframe id does not exist!,"status_id":6}
```