
CDPAY invoice appendix with API calls

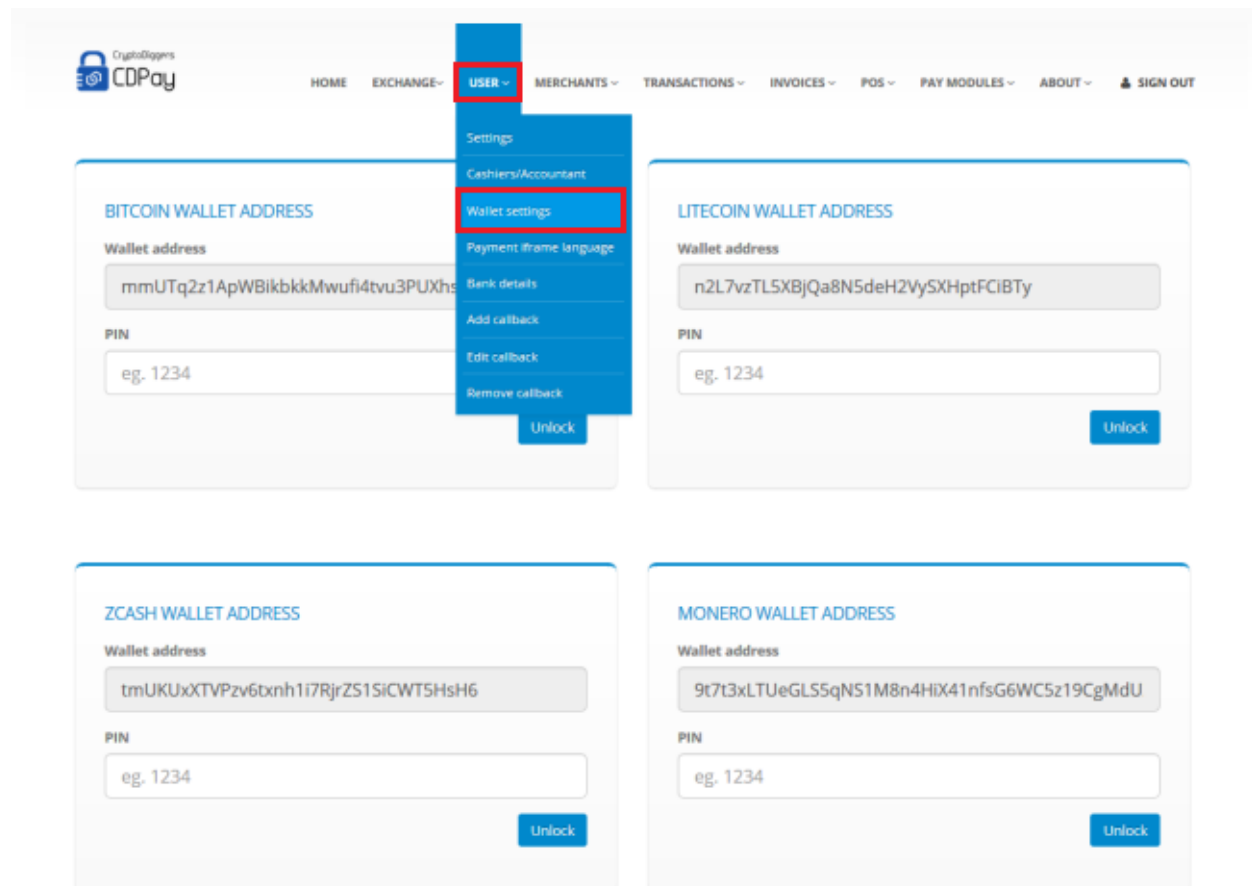


Intro

The documentation describes calls for the automated creation of an invoice appendix in the CDPAY system. The invoice appendix can be used as an addendum to a regular invoice to help view the QR code (s) for all currencies that the user account has set in the CDPAY system under its profile.

User profile setup

For the proper functioning of the invoice appendix, the user must have set up his cryptocurrency address (es) that he / she wants to accept. For users using the exchange to EUR / CZK, we set all addresses (unless otherwise agreed).





Accountant account creation

For companies requiring manually creating invoice appendix, a limited-account account with the following data, including its email, must be created for the accountant manager. Registered email will be used to obtain PDFs of invoice appendix document. The accountant will be able to sign into the CDPAY with her/his own account and create invoice appendix manually.

The screenshot displays the CDPay user management interface. At the top, a blue banner shows actual exchange rates for various cryptocurrencies. Below this, the main navigation bar includes 'HOME', 'EXCHANGE', 'USER', 'MERCHANTS', 'TRANSACTIONS', 'INVOICES', 'POS', 'PAY MODULES', 'ABOUT', and 'SIGN OUT'. The 'USER' menu is expanded, showing options like 'Add user', 'Maintain user', 'Edit user', 'Wallets for', 'Settings', 'Cashiers/Accountant', 'Wallet settings', 'Payment iframe language', 'Bank details', 'Add callback', 'Edit callback', and 'Remove callback'. The 'Add user' option is highlighted with a red box. The 'CREATE CASHIER / ACCOUNTANT USER' form is visible, containing fields for Username, Email, Password, Confirm password, Access role (set to 'Accountant'), and PIN. A 'Create' button is located at the bottom right of the form. To the right of the form, there is a section titled 'What does it mean additional users?' and 'WHY AND WHEN SHOULD I CREATE ADDITIONAL USERS?' with explanatory text and a list of roles: 'Teller' and 'Teller POS only'. A 'Remark' is also present at the bottom of this section.



Login as accountant user

Once you've created your account, you'll be able to sign in to the account immediately and begin creating invoice appendix.

rates: 3530.4 (EUR/BTC)28.77 (EUR/LTC)76.601 (EUR/DASH)57.532 (EUR/ZEC)47.08 (EUR/XMR)

CryptoDiggers CDPay

HOME PAY MODULES ~ ABOUT ~ SIGN IN

I'M A RETURNING CUSTOMER

Username

Password [\(Lost Password?\)](#)

Calculate formula and enter answer below: $16 + 1 =$

REGISTER AN ACCOUNT

Username

E-mail Address

Password Re-enter Password

PIN

Calculate formula and enter answer below: $3 + 24 =$

I agree to the [terms & conditions](#) of this website.



Manual invoice appendix creation

Manually creating an invoice appendix consists of your order number / invoice number / variable symbol, sums in FIAT currency (e.g. EUR), FIAT currencies in which you charge the company (for selection are EUR, USD, GBP, JPY, CNY, AUD, CAD, AED, CZK, PLZ, VND, CHF, NOK) and invoice expiration respectively validity of the QR code for payment. After this date, you will not be able to pay your invoice via CDPAY. Set it up according to your own needs.

The screenshot displays the CDPay web interface. At the top, a blue banner shows exchange rates: Actual exchange rates: 3530.4 (EUR/BTC)28.77 (EUR/LTC)76.601 (EUR/DASH)57.532 (EUR/ZEC)47.08 (EUR/XMR). The navigation bar includes HOME, EXCHANGE, INVOICES (highlighted with a red box), PAY MODULES, ABOUT, and SIGN OUT. A dropdown menu under INVOICES shows 'Create invoice' (highlighted with a red box) and 'Show invoices'. The main content area features a 'CREATE INVOICE' form (highlighted with a red box) with the following fields:

- Order ID (variable symbol): Enter Order ID / Variable symbol
- Amount in local currency: Amount in FIAT currency(e.g. 10.50)
- Choose invoice fiat currency: EUR
- Invoice expiration date: [Empty field]

A 'Create invoice' button is located at the bottom right of the form.



Invoice appendix email attachment

After you manually create the invoice appendix, the accountant receives the following email with a PDF attachment that you can attach or forward to your client.



Dear kalboni,

This email contains the invoice appendix attachment to allow your customers to pay via CDPay system with cryptocurrencies.

This email is generated automatically, please do not reply to it. If you need support, please contact us via contact form on our page or helpdesk <https://cryptodiggers.freshdesk.com>

Your CryptoDiggers team



Attachment PDF content

The QR code allows your end user to make a payment when it is the most advantageous for you during the validity of the invoice that you define in the call. The image is a preview and comes from a manually created invoice to a PDF file and an email account.



Invoice appendix

(valid until 2018-11-30)

buggie04

Bitcoin



Litecoin



Zcash



Monero





The URL will then link it to the page with the payment itself in the selected cryptocurrency.





CDPAY environments

Test: <https://www.cryptodiggerstest.eu/api/>

Live: <https://www.cdpay.eu/>

For the REST API call, the parameters must be called exactly in this order:

invoice.php

?a=create_invoice

&order_id=TestovaciOrder8

¤cy=1

&amount=10.25

&valid_date=2018-11-30

&api-sign

=ejZ38gAtuZh2AK%2Bdch1GQhBfFiBuXEKToeB5%2FJHOzGLUxojrHyNcRDsSSEmMfhb6OOyN4UJRiDqQYF
AMExFcNg%3D%3D

Example how to create signature:

```
$apikey=Your API key';
```

```
$seckey='Your SecKey';
```

```
$inputs["order_id"]= TestOrder8';
```

```
$inputs["amount"]=10.25;
```

```
$inputs["currency"]=1;
```

```
$inputs["api-key"]=$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;
```



PHP example

1. You need the PHP on your server side.
2. Download PHP library from CDPAY: https://www.cdpay.eu/client/cdpay_v2_class_php.tar.gz
3. Add to your code „require_once '<path_to_library>/cdpay2.php'”
4. Initialize class: „\$cdp=new cdpay(\$apiKey,\$apiSecret,true);”,
 - a. \$apiKey – your profile apikey from CDPAY.eu portal
 - b. \$apiSecret – your profile seckey from CDPAY.eu portal
 - c. True – if you want to access to test version <https://www.cryptodiggerstest.eu/api>
False – if you want to access production environment <https://www.cdpay.eu/>
5. You need to call the class as follows:

```
$tmp=$cdpay->requestCDPay(array('method'=>'create_invoice'),  
array('order_id'=>'TestovaciOrder8','currency'=>1,'amount'=>10.25,'valid_date'=>'2018-11-30'));
```

You have to change the parameters only in the second field

- a. order_id – your Order ID / Variable Symbol / some unique ID
- b. currency – fiat code number (EUR=1, USD=2, CZK=16, GBP=3, CAD=4, AUD=5, JPY=9, CNY=13, AED=17, PLZ=18, CHF=23, NOK=25)
- c. amount – amount where „.” separator is used
- d. valid_date – invoice expiration / URL QR code validity, after this date you won't be able to create payment with CDPAY portal.



All answers from CDPAY payment gateway are returned as JSON string.

In case of correct answer following message will be returned.

```
/var/www/cryptodiggers.eu/api/cdpay2.php:223:
array (size=3)
  'payments' =>
    array (size=5)
      'bcash' =>
        array (size=4)
          'ID' => int 24
          'currency_name' => string 'BCH' (length=3)
          'currency_text' => string 'bcash' (length=5)
          'payment_link' => string 'https://www.cryptodiggerstest.eu/api/invoice.php?invoiceid=EAB4409BEE4A1A90328C488EF1DD24A7C29AD128&crypto_id=24' (length=113)
      'bitcoin' =>
        array (size=4)
          'ID' => int 6
          'currency_name' => string 'BTC' (length=3)
          'currency_text' => string 'bitcoin' (length=7)
          'payment_link' => string 'https://www.cryptodiggerstest.eu/api/invoice.php?invoiceid=EAB4409BEE4A1A90328C488EF1DD24A7C29AD128&crypto_id=6' (length=112)
      'litecoin' =>
        array (size=4)
          'ID' => int 8
          'currency_name' => string 'LTC' (length=3)
          'currency_text' => string 'litecoin' (length=8)
          'payment_link' => string 'https://www.cryptodiggerstest.eu/api/invoice.php?invoiceid=EAB4409BEE4A1A90328C488EF1DD24A7C29AD128&crypto_id=8' (length=112)
      'monero' =>
        array (size=4)
          'ID' => int 20
          'currency_name' => string 'XMR' (length=3)
          'currency_text' => string 'monero' (length=6)
          'payment_link' => string 'https://www.cryptodiggerstest.eu/api/invoice.php?invoiceid=EAB4409BEE4A1A90328C488EF1DD24A7C29AD128&crypto_id=20' (length=113)
      'zcash' =>
        array (size=4)
          'ID' => int 21
          'currency_name' => string 'ZEC' (length=3)
          'currency_text' => string 'zcash' (length=5)
          'payment_link' => string 'https://www.cryptodiggerstest.eu/api/invoice.php?invoiceid=EAB4409BEE4A1A90328C488EF1DD24A7C29AD128&crypto_id=21' (length=113)
  'error' => int 0
  'error_msg' => string '' (length=0)
```

In case of wrong answer following message will be returned:

```
/var/www/cryptodiggers.eu/api/cdpay2.php:223:
array (size=2)
  'error' => int 1
  'error_msg' => string 'Order ID already exists.' (length=24)
```