accounts@coinapult.com
coinapult.com

COINAPULT

Coinapult API Specification

 v1.9.0
Author: Ira Miller, Guilherme Polo 5/25/2014

Summary

Coinapult makes all core wallet functionality, and additional merchant tools available via API. The API is RESTful and all the returned messages are in JSON.

Authentication

Most API functions require authentication. To authenticate a request, the following steps should be followed

1. Include a unique 'nonce' argument
2. Include the current unix timestamp as 'timestamp'
3. Include an 'endpoint' argument set to the end point for the request
4. JSON encode the values and convert it to the base64 encoding, call it SIGNDATA
5. Create HMAC of SIGNDATA using your API secret and SHA512
6. Add HMAC to the request header field 'cpt-hmac'
7. Add API key to the request header field 'cpt-key'
8. Send a POST request with the single parameter 'data' set to SIGNDATA

Note that the timestamp must be less than 1 hour old, and for all purposes the nonce must be unique within an API key. If either of these is not the case, or the HMAC doesn't match expectations, an HTTP 401 error will be returned.

Errors

If any errors occur when handling an API request, the response will follow this format:

{“error”: “error message describing problem goes here”}

Any response that contains the 'error' field should be considered unsuccessful for the reason given. Note that HTTP errors may also be thrown depending on the issue.

Oceania Business Plaza
Torre 2000 15a
Panama City, Panama

accounts@coinapult.com
coinapult.com

COINAPULT

Warnings

Any API request may return an optional ‘warning’ field to describe some specific situation with a given request. The general format for a warning message is:

```
{
        “category”: “category for this warning”,
        “message”: “message about this warning”,
        … optional fields based on the category …
}
```

Currently there is a single category for warning messages: “deprecated”. For those messages there is a single optional field “alternative” where it’s value is a string that points to a route that supersedes the one used in the request.

Callbacks

If a callback URL is specified in the transaction, a callback will be made when the transaction enters a final state (complete, canceled). Each callback will contain a JSON encoded transaction record sent via POST. Here are two examples:

```
{
        "transaction_id":"52289f703b57f529911fe8e3",
        "timestamp":1394914912,
        "completeTime":1394915012,
        "quote":{
                "ask": 579.41,
                "bid": 571.85
        },
        "in":{
                "expected": 10.0,
                "currency": "BTC",
                "amount": 10.0
        },
        "state": "complete",
        "meta": [],
        "expiration": 1394915812,
        "out": {
                "expected": 5718.50,
                "currency": "USD",
                "amount": 5718.50
        },
        "type": "invoice",
        "address": "1FvDD4YqVac9Zk7FHrD5EkkJdMBJVeh4ho",
},
{
        "transaction_id": "52fd3a146f2cc034937a66ae",
```

Oceania Business Plaza
Torre 2000 15a
Panama City, Panama

accounts@coinapult.com
coinapult.com


```
  "timestamp": 1363382104,
"completeTime":None,
"in": {
        "expected": 14.15,
        "currency": “BTC”,
        "amount": 0.0
},
"state": "canceled",
"meta": [],
"out": {
        "expected": 14.15,
        "currency": "BTC",
        "amount": 0.0
}
"type": "payment"
}
```


If there are any errors, the callback will include an 'errors' list, describing the problems encountered, and the state should be 'canceled.' We will sign each request with your key and generate an HMAC using your secret. To authenticate the callback, follow these steps:

1. Compare API key to header field 'cpt-key'
2. Create HMAC for the 'data' in the POST request using your API secret and SHA512
3. Compare generated HMAC to header field 'cpt-hmac'

We will not include a nonce in the callback, but will send a timestamp. After authenticating the callback, decode 'data' from base64 and the resulting content will be a transaction JSON encoded.

Merchant Flow Example

Typically, a merchant wishing to accept payment will only need two functions: Create Invoice and Search. The flow would go as such:

1. User submits order on merchant site
2. Merchant makes Receive call to Coinapult and saves the returned transaction_id
3. Merchant directs customer to /check/invoice/<transaction_id>
4. Customer pays invoice using Bitcoin network or pre-existing balance

Oceania Business Plaza
Torre 2000 15a
Panama City, Panama

accounts@coinapult.com
coinapult.com

COINAPULT

Private API Functions

All the calls to private functions need to be signed and sent as POST requests.

Receive - [/api/t/receive]

params:

- timestamp (UTC, unix style)
- nonce (string)
- endpoint (/t/receive)
- currency (BTC)
- outCurrency (BTC, USD) (optional)
- amount (float) (optional)
- outAmount (float) (optional)
- callback (url)(optional)

return:

- transaction (json) – A scrubbed transaction as in the above examples

The default action is to specify the input amount and currency. By also using the outAmount and outCurrency fields, conversion transactions can be made.

For instance, a transaction with currency=BTC, outCurrency=USD, and outAmount=100 would result in an invoice to be paid in bitcoins, and the recipient is to receive $100. The system will create a quote at current market rate, locking in the price for 15 minutes. The response will follow the same format as a callback. The amount to be paid is the response['in']['expected'] amount.

Send - [/api/t/send]

Send a specified amount to a given address

params:

- timestamp (UTC, unix style)
- nonce (string)
- endpoint (/t/send)
- currency (BTC, USD)
- amount (float)
- outAmount (float)
- address (recipient's address)
- callback (url)(optional)

return:

- transaction (json) – A scrubbed transaction as in the above examples

Oceania Business Plaza
Torre 2000 15a
Panama City, Panama

accounts@coinapult.com
coinapult.com

COINAPULT

Convert - [/api/t/convert]
Convert an amount between two currencies

params:
- timestamp (UTC, unix style)
- nonce (string)
- endpoint (/t/convert)
- inCurrency (BTC, USD)
- outCurrency (BTC, USD)
- amount (of in currency)
- callback (url) (optional)

return:
- transaction (json) – A scrubbed transaction as in the above examples

Search - [/api/t/search]
Search transaction history by common keys

params:
- timestamp (UTC, unix style)
- nonce (string)
- endpoint (/t/search)
- transaction_id (optional)
- type (invoice, payment, conversion, etc.) (optional)
- currency (BTC, USD) (optional)
- to (optional)
- from (optional)
- txhash (string) (optional)
- many (boolean) (optional)
- page (int) (optional)

return:
- One or more scrubbed transactions

By default, transaction search is restricted to returning a single transaction that matches the specified criteria according to the combination of ‘transaction_id’, ‘type’, ‘currency’, ‘to’, ‘from’, and ‘txhash’ parameters. The last one refers to the txid returned by certain bitcoind calls.

To search for multiple transactions, set the ‘many’ field to ‘1’. When searching for multiple transactions, the response will include a ‘pageCount’ field determining how many other pages of results you can look for, and also a ‘results’ field with a list of the returned transactions. To specify a page, set the ‘page’ field to the desired integer number.

Oceania Business Plaza
Torre 2000 15a
Panama City, Panama

accounts@coinapult.com
coinapult.com

COINAPULT

Lock – [/api/t/lock]
Lock a certain amount of bitcoins to another currency.

params:
- timestamp (UTC, unix style)
- nonce (string)
- endpoint (/t/lock)
- amount (float)
- outAmount (float) (optional)
- currency (USD) (optional)
- callback (url) (optional)

return:
- transaction (json) – A scrubbed transaction as in the above examples

The default action is to lock a certain amount of bitcoins to USD. Alternatively, outAmount in the given currency might be specified and then the system will figure out the corresponding amount of bitcoins that should be locked. It is not allowed to specify both amount and outAmount.

Unlock – [/api/t/unlock]
Unlock a certain amount in a given currency to get bitcoins back.

params:
- timestamp (UTC, unix style)
- nonce (string)
- endpoint (/t/unlock)
- amount (float or ‘all’)
- address (recipient’s bitcoin address)
- outAmount (float or ‘all’) (optional)
- currency (USD) (optional)
- callback (url) (optional)

return:
- transaction (json) – A scrubbed transaction as in the above examples

To unlock a previously created lock, specify a bitcoin address to where the coins must be sent to. In this case, amount refers to the USD amount in the locks, while outAmount is the amount in bitcoins. It is not allowed to specify both amount and outAmount.

Oceania Business Plaza
Torre 2000 15a
Panama City, Panama

accounts@coinapult.com
coinapult.com

COINAPULT

Get Bitcoin Address - [/api/getbitcoinaddress]
Get a new receiving Bitcoin address for your account.

params:
- timestamp (UTC, unix style)
- nonce (string)
- endpoint (/getbitcoinaddress)

return:
- address – a bitcoin address for your account

Historically, this request returns a single string as the result therefore it is not JSON. If you prefer to get a JSON here, use /api/getBitcoinAddress with the same endpoint parameter. The return value will be in JSON with a single key "address" where its value is the bitcoin address for your account.

Account Information – [/api/accountInfo]

params:
- timestamp (UTC, unix style)
- nonce (string)
- endpoint (/accountInfo)
- balanceType ('all', 'locks', 'normal') (optional)
- locksAsBTC (boolean) (optional)
- outAmount (float or 'all') (optional)

return:
- role: (string)
- balances:

For each currency in the account's wallet
  - amount: (float)
  - currency: (string)

Oceania Business Plaza
Torre 2000 15a
Panama City, Panama

accounts@coinapult.com
coinapult.com

COINAPULT

Public API Functions

All the calls to public functions need to be sent as GET requests.

Get exchange rates - [/api/getRates] *deprecated*
Get recommended retail exchange rates of all supported pairs.

params:
return:

- (timestamp)
  - ask
    - (int)
    - (float)
  - bid
    - (int)
    - (float)

Ticker – [/api/ticker]
Get recommended retail exchange rates of all supported pairs at different levels.

params:

- begin (UTC, unix style) (optional)
- end (UTC, unix style) (optional)

return:
If `begin' and `end' are not specified, then

- updatetime: timestamp (UTC, unix style)
- market: (string)
- index: (float)

For each key in small, medium, large, vip, vip+, 100, 500, 2000, 5000, 10000

- ask: (float)
- bid: (float)
-

If `begin' or `end' are specified, then a JSON object of n items where each item is composed by

- updatetime: timestamp (UTC, unix style)
- ask: (float)
- bid: (float)

Oceania Business Plaza
Torre 2000 15a
Panama City, Panama