

COINIGY REST API DOCUMENTATION

UPDATED: 7/14/2015 8:16 PM

ENDPOINT: <https://www.coinigy.com/api/v1>

RETURN FORMAT: JSON

ALLOWED METHODS: POST

ALL PARAMETERS MUST BE POST PARAMETERS
GET/DELETE/PUT PARAMETERS ARE NOT ALLOWED

DATA METHODS

[accounts](#)
[activity](#)
[alerts](#)
[balances](#)
[data \(history\)](#)
[data \(asks\)](#)
[data \(bids\)](#)
[data \(orders\)](#)
[data \(all\)](#)
[exchanges](#)
[markets](#)
[newsFeed](#)
[orders](#)
[orderTypes](#)
[pushNotifications](#)

ACTION METHODS

[addAlert](#)
[addOrder](#)
[cancelOrder](#)
[deleteAlert](#)
[refreshBalance](#)

OTHER TOPICS

[Authentication](#)
[Notifications](#)
[Errors](#)

AUTHENTICATION

Private key authentication is used to authenticate to the Coinigy API. You can generate an API key and secret code under Settings -> My Account when you log into your Coinigy account. This key and secret must be included in the header of all API requests. All requests must be made over HTTPS.

```
$headers = array(  
    'X-API-KEY: abcdefghijklmnopqrstuvwxyz123456',  
    'X-API-SECRET: 654321zyxwvutsrqponmlkjihgfedcba'  
);
```

[BACK TO TOP](#)

NOTIFICATIONS

The API may at any time return a [notification] node in the return data. This prevents your app from ever needing to create, trigger or determine messages about the result of an action. As long as a process is in place to always show these notifications if they are present, the user will be informed of the results of all their actions. In this way, all the user's actions are also archived in the database for reviewing or replaying and left consistent on all devices.

The format of the notification object has some important aspects to know in order to properly display the notification.

The two fields `notification_type_title` and `notification_type_message` contain the title and message for the notification.

Within the values of these variables you may find tokens like `%1$s` or `%2$s`. These are placeholders into which you must inject the values stored in `notification_title_vars` and `notification_vars`.

These two `_var` fields contain a CSV string that should be injected into the placeholder tokens. The index position of the value in the CSV string corresponds to the numeric value within the placeholder (ie. `%1$s` value comes from the first CSV value and `%2$s` comes from the 2nd, etc).

```
[notifications] => Array
(
  [0] => stdClass Object
  (
    [notification_id] => 263344
    [notification_type_title] => Alert Request Added at %1$s
    [notification_type_message] => Request for [%1$s] %2$s @ %3$s added
    [notification_style] => success
    [notification_vars] => OK,BTC/CNY,650
    [notification_title_vars] => 2014-10-18 10:29:17
    [notification_pinned] => 0
    [notification_sound] => 0
    [notification_sound_id] => alertSound
  )
)
```

Java/C# `String.Format()` method should get you started in the right direction for injecting the "vars" fields into the title and message fields:

<http://docs.oracle.com/javase/6/docs/api/java/util/Formatter.html>

[https://msdn.microsoft.com/en-us/library/system.string.format\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.string.format(v=vs.110).aspx)

You can use `sprintf()` in PHP:

<http://php.net/sprintf>

[BACK TO TOP](#)

| | |
|--------------|--|
| METHOD ID: 5 | accounts (List of your attached exchange API keys) |
| PARAMETERS | none |
| RETURN DATA | <pre>stdClass Object ([data] => Array ([0] => stdClass Object ([auth_id] => 27 [auth_key] => 81095f14272cb3afa72530b [auth_optional1] => [auth_nickname] => Nickname [exch_name] => Bitcoin Exchange [auth_secret] => ***** [auth_active] => 1 [auth_trade] => 1 [exch_trade_enabled] => 1))))</pre> |

[BACK TO TOP](#)

| | |
|--------------|---|
| METHOD ID: 6 | activity (List of your recent activity) |
| PARAMETERS | none |
| RETURN DATA | <pre>stdClass Object ([data] => Array ([0] => stdClass Object ([notification_id] => 263031 [notification_type_title] => Price Alert %s [notification_type_message] => %1\$s [%2\$s] @ %3\$s [notification_style] => info [notification_vars] => MCX\BTC,MCX,.03350101 [notification_title_vars] => 2014-09-25 21:42:28 [notification_pinned] => 1 [notification_sound] => 1 [notification_sound_id] => alertSound))))</pre> |

[BACK TO TOP](#)

| | |
|--------------|---|
| METHOD ID: 7 | balances (List of all current balances for an account) |
| PARAMETERS | <ol style="list-style-type: none">1. show_nils 0 or 1 to show empty balances2. auth_ids CSV (, or _) of auth_ids for which to return data |
| RETURN DATA | <pre>stdClass Object ([data] => Array ([0] => stdClass Object ([balance_curr_code] => BTC [balance_amount_avail] => 0.1 [balance_amount_held] => 0.001 [balance_amount_total] => 0.101 [btc_balance] => 0.101 [last_price] => 500.00)))</pre> |

[BACK TO TOP](#)

| | |
|--------------|--|
| METHOD ID: 9 | pushNotifications (List any unshown alerts or trade notifications) |
| PARAMETERS | none |
| RETURN DATA | <pre>stdClass Object ([data] => Array ([0] => stdClass Object ([notification_id] => 263031 [notification_type_title] => Price Alert %s [notification_type_message] => %1\$s [%2\$s] @ %3\$s [notification_style] => info [notification_vars] => MCX\ /BTC,MCX, .03350101 [notification_title_vars] => 2014-09-25 21:42:28 [notification_pinned] => 1 [notification_sound] => 1 [notification_sound_id] => alertSound)))</pre> |

[BACK TO TOP](#)

| | |
|---------------|--|
| METHOD ID: 10 | orders (List of all open user orders and order history) |
| PARAMETERS | none |
| RETURN DATA | <pre>stdClass Object ([data] => stdClass Object ([open_orders] => Array ([0] => stdClass Object ([exch_name] => Cryptsy [mkt_name] => LTC/BTC [limit_price] => 0.01 [operator] => [order_id] => 2582 [order_type] => Buy [order_price_type] => Limit [order_status] => Open [quantity] => 0.1 [order_time] => 2014-09-22 [foreign_order_id] => 154026995 [auth_nickname] => Nickname [auth_id] => 27 [quantity_remaining] => 0.1)) [order_history] => Array ([0] => stdClass Object ([exch_id] => 1 [exch_name] => Cryptsy [mkt_name] => LTC/BTC [auth_id] => 27 [executed_price] => 0.005 [limit_price] => 0.005 [order_id] => 883 [order_type] => Buy [order_price_type] => Limit [order_status] => Executed [quantity] => 0.2 [quantity_remaining] => 0.2 [last_updated] => 2014-08-19 [order_time] => 2014-08-19 [auth_nickname] => Nickname))))</pre> |

[BACK TO TOP](#)

| | |
|---------------|--|
| METHOD ID: 11 | alerts (List of all active alerts and alert history) |
| PARAMETERS | none |
| RETURN DATA | <pre>stdClass Object ([data] => stdClass Object ([open_alerts] => Array ([0] => stdClass Object ([exch_name] => BTC-e [mkt_name] => BTC/USD [price] => 650.0000000000 [operator] => > [alert_id] => 1322 [operator_text] => Rises above 650.)) [alert_history] => Array ([0] => stdClass Object ([exch_name] => mcxNOW [mkt_name] => MCX/BTC [price] => 0.1770000000 [operator] => < [alert_history_id] => 368 [timestamp] => 2014-03-25 17:20:17 [alert_price] => 0.173 [operator_text] => Dropped below 0.177)))</pre> |

[BACK TO TOP](#)

| | |
|---------------|--|
| METHOD ID: 12 | exchanges (List of all active exchanges) |
| PARAMETERS | none |
| RETURN DATA | <pre>stdClass Object ([data] => Array ([0] => stdClass Object ([exch_id] => 1 [exch_name] => Cryptsy [exch_code] => CRYP [exch_fee] => 0.003 [exch_trade_enabled] => 1 [exch_balance_enabled] => 1 [exch_url] =>)))</pre> |

[BACK TO TOP](#)

| | |
|---------------|--|
| METHOD ID: 13 | markets (List of all markets for an exchange) |
| PARAMETERS | 1. exchange_code Filter markets returned by exchange (optional) |
| RETURN DATA | <pre>stdClass Object ([data] => Array ([0] => stdClass Object ([exch_id] => 8 [exch_name] => BTC China [exch_code] => BTCC [mkt_id] => 45 [mkt_name] => CNY/BTC [exchmkt_id] => 123)))</pre> |

[BACK TO TOP](#)

| | |
|---------------|--|
| METHOD ID: 14 | data {type: history} (Recent trade history for a market) |
| PARAMETERS | <ol style="list-style-type: none">1. exchange_code Exchange for which you want data2. exchange_market Market for which you want data3. type Data type to retrieve ("history") |
| RETURN DATA | <pre>stdClass Object ([data] => stdClass Object ([primary_curr_code] => BTC [secondary_curr_code] => CNY [type] => history [history] => Array ([0] => stdClass Object ([price] => 2520.2000000000 [quantity] => 0.1400000000 [time_local] => 2014-09-26 [type] => SELL))))</pre> |

[BACK TO TOP](#)

| | |
|---------------|--|
| METHOD ID: 14 | data {type: asks} (Open sell orders for a market) |
| PARAMETERS | <ol style="list-style-type: none">1. exchange_code Exchange for which you want data2. exchange_market Market for which you want data3. type Data type to retrieve ("asks") |
| RETURN DATA | <pre>stdClass Object ([data] => stdClass Object ([primary_curr_code] => BTC [secondary_curr_code] => CNY [type] => asks [asks] => Array ([0] => stdClass Object ([price] => 2604.1700000000 [quantity] => 13.6400000000 [total] => 35520.8788000000))))</pre> |

[BACK TO TOP](#)

| | |
|---------------|--|
| METHOD ID: 14 | data {type: bids} (Open buy orders for a market) |
| PARAMETERS | <ol style="list-style-type: none">1. exchange_code Exchange for which you want data2. exchange_market Market for which you want data3. type Data type to retrieve ("bids") |
| RETURN DATA | <pre>stdClass Object ([data] => stdClass Object ([primary_curr_code] => BTC [secondary_curr_code] => CNY [type] => bids [bids] => Array ([0] => stdClass Object ([price] => 2604.1700000000 [quantity] => 13.6400000000 [total] => 35520.8788000000))))</pre> |

[BACK TO TOP](#)

| | |
|---------------|---|
| METHOD ID: 14 | data {type: orders} (Bids and Asks) |
| PARAMETERS | <ol style="list-style-type: none">1. exchange_code Exchange for which you want data2. exchange_market Market for which you want data3. type Data type to retrieve ("orders") |
| RETURN DATA | <pre>stdClass Object ([data] => stdClass Object ([primary_curr_code] => BTC [secondary_curr_code] => CNY [type] => orders [bids] => Array ([0] => stdClass Object ([price] => 2604.1700000000 [quantity] => 13.6400000000 [total] => 35520.8788000000)) [asks] => Array ([0] => stdClass Object ([price] => 2604.1700000000 [quantity] => 13.6400000000 [total] => 35520.8788000000))))</pre> |

[BACK TO TOP](#)

| | |
|---------------|---|
| METHOD ID: 14 | data {type: all} (History, Bids and Asks) |
| PARAMETERS | <ol style="list-style-type: none">1. exchange_code Exchange for which you want data2. exchange_market Market for which you want data3. type Data type to retrieve ("all") |
| RETURN DATA | <pre>stdClass Object ([data] => stdClass Object ([primary_curr_code] => BTC [secondary_curr_code] => CNY [type] => all [history] => Array ([0] => stdClass Object ([price] => 2520.2000000000 [quantity] => 0.1400000000 [time_local] => 2014-09-26 01:37:12 [type] => SELL)) [bids] => Array ([0] => stdClass Object ([price] => 2604.1700000000 [quantity] => 13.6400000000 [total] => 35520.8788000000)) [asks] => Array ([0] => stdClass Object ([price] => 2604.1700000000 [quantity] => 13.6400000000 [total] => 35520.8788000000))))</pre> |

[BACK TO TOP](#)

| | |
|---------------|---|
| METHOD ID: 23 | newsFeed (Get site's list of current news articles) |
| PARAMETERS | none |
| RETURN DATA | <pre>stdClass Object ([data] => Array ([0] => stdClass Object ([id] => 3952462 [url] => https://www.cr [title] => U.S. Lawmakers [pubDate] => 6-Oct-14 [timestamp] => 2014-10-06 21:11:00 [feed_id] => 1 [published_date] => 2014-10-06 21:00:00 [feed_name] => Cryptocoin News [feed_url] => http://www.cryptoco [feed_enabled] => 1 [feed_description] => News [url_field] => link [title_field] => title [date_field] => pubDate [feed_image] => ccnews.png))))</pre> |

[BACK TO TOP](#)

| | |
|---------------|---|
| METHOD ID: 35 | orderTypes (Get order type and price type used for placing orders) |
| PARAMETERS | none |
| RETURN DATA | <pre>stdClass Object ([data] => stdClass Object ([order_types] => Array ([0] => stdClass Object ([order_type_id] => 1 [name] => Buy [order] => 5) [1] => stdClass Object ([order_type_id] => 2 [name] => Sell [order] => 10)) [price_types] => Array ([0] => stdClass Object ([price_type_id] => 3 [name] => Limit [order] => 15) [1] => stdClass Object ([price_type_id] => 6 [name] => Stop Limit [order] => 30))))</pre> |

[BACK TO TOP](#)

| | |
|---------------|--|
| METHOD ID: 23 | refreshBalance (Refresh a users's account balances from server) |
| PARAMETERS | none |
| RETURN DATA | <pre>stdClass Object ([data] => Array ([0] => stdClass Object ([balance_curr_code] => BTC [balance_amount_avail] => 1.5 [balance_amount_held] => 0 [balance_amount_total] => 1.5 [btc_balance] => 1.5 [last_price] => 0) [1] => stdClass Object ([balance_curr_code] => USD [balance_amount_avail] => 10 [balance_amount_held] => 0 [balance_amount_total] => 10 [btc_balance] => 0.0287 [last_price] => 287)))</pre> |

[BACK TO TOP](#)

| | |
|---------------|---|
| METHOD ID: 31 | addAlert (Add a new price alert) |
| PARAMETERS | <ol style="list-style-type: none">1. <code>exch_code</code> 4 character exchange code2. <code>market_name</code> Name of market3. <code>alert_price</code> Price to set alert at4. <code>alert_note</code> Custom note text from user. |
| RETURN DATA | { } |

[BACK TO TOP](#)

| | |
|---------------|---|
| METHOD ID: 36 | deleteAlert (Delete a price alert) |
| PARAMETERS | <ol style="list-style-type: none">1. <code>alert_id</code> Numeric ID for alert to delete |
| RETURN DATA | { } |

[BACK TO TOP](#)

| | |
|---------------|--|
| METHOD ID: 33 | addOrder (Add a new exchange order) |
| PARAMETERS | <ol style="list-style-type: none">1. <code>auth_id</code> Numeric ID for API Account2. <code>exch_id</code> ID of Exchange3. <code>mkt_id</code> ID of Market4. <code>order_type_id</code> Order type from orderTypes API call5. <code>price_type_id</code> Price type from orderTypes API call6. <code>limit_price</code> Buy/Sell Price7. <code>stop_price</code> Price to trigger a stop price type8. <code>order_quantity</code> Quantity for order |
| RETURN DATA | { } |

[BACK TO TOP](#)

| | |
|---------------|---|
| METHOD ID: 34 | cancelOrder (Cancel an exchange order) |
| PARAMETERS | <ol style="list-style-type: none">1. <code>internal_order_id</code> Numeric ID for order to delete (not foreign order id) |
| RETURN DATA | { } |

[BACK TO TOP](#)

ERRORS

If your root level return object has an [error] OR [err_num] node then you have encountered an error of some kind. The error will have a message which should be displayed and an error number for debugging purposes.

ERR_NUM: 1055-00-01 == ERROR NUMBER – METHOD ID – INSTANCE #

```
stdClass Object
(
  [err_num] => 1000-10-01
  [err_msg] => Error Message.
)
```

1055 – INVALID API KEY

1056 – RETURN DATA NOT PROPERLY FORMATTED

1057 – POSTED PARAMETERS WERE EITHER MISSING OR NOT PROPERLY FORMATTED

1061 – TRY/CATCH EXCEPTION

1062 – NO API METHOD PROVIDED