

PHP Binance Api

Generated by Doxygen 1.8.6

Fri Mar 30 2018 15:14:45

Contents

1	Namespace Index	1
1.1	Namespace List	1
2	Class Index	3
2.1	Class List	3
3	Namespace Documentation	5
3.1	Binance Namespace Reference	5
4	Class Documentation	7
4.1	Binance\API Class Reference	7
4.1.1	Detailed Description	9
4.1.2	Constructor & Destructor Documentation	10
4.1.2.1	__construct	10
4.1.3	Member Function Documentation	10
4.1.3.1	account	10
4.1.3.2	addToTransferred	10
4.1.3.3	aggTrades	10
4.1.3.4	balances	11
4.1.3.5	bookPrices	11
4.1.3.6	buy	11
4.1.3.7	buyTest	12
4.1.3.8	cancel	12
4.1.3.9	candlesticks	12
4.1.3.10	chart	13
4.1.3.11	cumulative	13
4.1.3.12	depositAddress	13
4.1.3.13	depositHistory	14
4.1.3.14	depth	14
4.1.3.15	depthCache	14
4.1.3.16	displayDepth	15
4.1.3.17	exchangeInfo	15

4.1.3.18	first	15
4.1.3.19	getProxyUriString	15
4.1.3.20	getRequestCount	16
4.1.3.21	getTransferred	16
4.1.3.22	highstock	16
4.1.3.23	history	16
4.1.3.24	keepAlive	17
4.1.3.25	last	17
4.1.3.26	marketBuy	17
4.1.3.27	marketBuyTest	18
4.1.3.28	marketSell	18
4.1.3.29	marketSellTest	18
4.1.3.30	miniTicker	19
4.1.3.31	openOrders	20
4.1.3.32	order	20
4.1.3.33	orders	20
4.1.3.34	orderStatus	21
4.1.3.35	prevDay	21
4.1.3.36	prices	21
4.1.3.37	sell	22
4.1.3.38	sellTest	22
4.1.3.39	setProxy	22
4.1.3.40	sortDepth	23
4.1.3.41	ticker	23
4.1.3.42	time	23
4.1.3.43	trades	23
4.1.3.44	userData	24
4.1.3.45	useServerTime	24
4.1.3.46	withdraw	24
4.1.3.47	withdrawHistory	25
4.1.4	Member Data Documentation	25
4.1.4.1	\$api_key	25
4.1.4.2	\$api_secret	25
4.1.4.3	\$balances	25
4.1.4.4	\$base	25
4.1.4.5	\$btc_total	25
4.1.4.6	\$btc_value	25
4.1.4.7	\$chartQueue	26
4.1.4.8	\$charts	26
4.1.4.9	\$depthCache	26

4.1.4.10	\$depthQueue	26
4.1.4.11	\$httpDebug	26
4.1.4.12	\$info	26
4.1.4.13	\$proxyConf	26
4.1.4.14	\$requestCount	26
4.1.4.15	\$transferred	26
4.1.4.16	\$wapi	26
 Index		 27

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Binance	5
-----------------------------------	---

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

[Binance\API](#)

 Main Binance class [7](#)

Chapter 3

Namespace Documentation

3.1 Binance Namespace Reference

Classes

- class [API](#)

Main Binance class.

Chapter 4

Class Documentation

4.1 Binance\API Class Reference

Main Binance class.

Public Member Functions

- **__construct** (string \$api_key= "", string \$api_secret= "", array \$options=["useServerTime"=>false], array \$proxyConf=null)
Constructor for the class, There are 4 ways to construct the class:
- **buy** (string \$symbol, \$quantity, \$price, string \$type="LIMIT", array \$flags=[])
buy attempts to create a currency order each currency supports a number of order types, such as -LIMIT -MARKET -STOP_LOSS -STOP_LOSS_LIMIT -TAKE_PROFIT -TAKE_PROFIT_LIMIT -LIMIT_MAKER
- **buyTest** (string \$symbol, \$quantity, \$price, string \$type="LIMIT", array \$flags=[])
buyTest attempts to create a TEST currency order
- **sell** (string \$symbol, \$quantity, \$price, string \$type="LIMIT", array \$flags=[])
sell attempts to create a currency order each currency supports a number of order types, such as -LIMIT -MARKET -STOP_LOSS -STOP_LOSS_LIMIT -TAKE_PROFIT -TAKE_PROFIT_LIMIT -LIMIT_MAKER
- **sellTest** (string \$symbol, \$quantity, \$price, string \$type="LIMIT", array \$flags=[])
sellTest attempts to create a TEST currency order
- **marketBuy** (string \$symbol, \$quantity, array \$flags=[])
marketBuy attempts to create a currency order at given market price
- **marketBuyTest** (string \$symbol, \$quantity, array \$flags=[])
marketBuyTest attempts to create a TEST currency order at given market price
- **marketSell** (string \$symbol, \$quantity, array \$flags=[])
marketSell attempts to create a currency order at given market price
- **marketSellTest** (string \$symbol, \$quantity, array \$flags=[])
marketSellTest attempts to create a TEST currency order at given market price
- **cancel** (string \$symbol, \$orderid)
cancel attempts to cancel a currency order
- **orderStatus** (string \$symbol, \$orderid)
orderStatus attempts to get orders status
- **openOrders** (string \$symbol=null)
openOrders attempts to get open orders for all currencies or a specific currency
- **orders** (string \$symbol, int \$limit=500, int \$fromOrderId=1)
orders attempts to get the orders for all or a specific currency
- **history** (string \$symbol, int \$limit=500, int \$fromTradeId=1)

- history* Get the complete account trade history for all or a specific currency
- [useServerTime](#) ()
 - useServerTime* adds the 'useServerTime'=>true to the API request to avoid time errors
- [time](#) ()
 - time* Gets the server time
- [exchangeInfo](#) ()
 - exchangeInfo* Gets the complete exchange info, including limits, currency options etc.
- [withdraw](#) (string \$asset, string \$address, \$amount, \$addressTag=null)
 - withdraw* requests a asset be withdrawn from binance to another wallet
- [depositAddress](#) (string \$asset)
 - depositAddress* get the deposit address for an asset
- [depositHistory](#) (string \$asset=null)
 - depositAddress* get the deposit history for an asset
- [withdrawHistory](#) (string \$asset=null)
 - withdrawHistory* get the withdrawal history for an asset
- [prices](#) ()
 - prices* get all the current prices
- [bookPrices](#) ()
 - bookPrices* get all bid/asks prices
- [account](#) ()
 - account* get all information about the api account
- [prevDay](#) (string \$symbol)
 - prevDay* get 24hr ticker price change statistics for symbols
- [aggTrades](#) (string \$symbol)
 - aggTrades* get Market History / Aggregate Trades
- [depth](#) (string \$symbol)
 - depth* get Market depth
- [balances](#) (\$priceData=false)
 - balances* get balances for the account assets
- [getProxyUriString](#) ()
 - getProxyUriString* get Uniform Resource Identifier string assocaited with proxy config
- [setProxy](#) (array \$proxyconf)
 - setProxy* set proxy config by passing in an array of the proxy configuration
- [order](#) (string \$side, string \$symbol, \$quantity, \$price, string \$type="LIMIT", array \$flags=[], bool \$test=false)
 - order* formats the orders before sending them to the curl wrapper function You can call this function directly or use the helper functions
- [candlesticks](#) (string \$symbol, string \$interval="5m", int \$limit=null, \$startTime=null, \$endTime=null)
 - candlesticks* get the candel for the given intervals 1m,3m,5m,15m,30m,1h,2h,4h,6h,8h,12h,1d,3d,1w,1M
- [cumulative](#) (array \$depth)
 - cumulative* Converts depth cache into a cumulative array
- [highstock](#) (array \$chart, bool \$include_volume=false)
 - highstock* Converts Chart Data into array for highstock & kline charts
- [first](#) (array \$array)
 - first* Gets first key of an array
- [last](#) (array \$array)
 - last* Gets last key of an array
- [displayDepth](#) (array \$array)
 - displayDepth* Formats nicely for console output
- [getTransferred](#) ()
 - getTransferred* gets the total transfered in b,Kb,Mb,Gb
- [getRequestCount](#) ()

- getRequestCount gets the total number of API calls*

 - **addToTransferred** (int \$int)

addToTransferred add interger bytes to the total transfered also incrementes the api counter
- **sortDepth** (string \$symbol, int \$limit=11)

sortDepth Sorts depth data for display & getting highest bid and lowest ask
- **depthCache** (\$symbols, Callable \$callback)

depthCache Pulls /depth data and subscribes to WebSocket endpoint Maintains a local Depth Cache in sync via lastUpdated.
- **trades** (\$symbols, Callable \$callback)

trades Trades WebSocket Endpoint
- **ticker** (\$symbol, Callable \$callback)

ticker pulls 24h price change statistics via WebSocket
- **chart** (\$symbols, string \$interval="30m", Callable \$callback)

chart Pulls /kline data and subscribes to WebSocket endpoint
- **keepAlive** ()

keepAlive Keep-alive function for userDataStream
- **userData** (&\$balance_callback, &\$execution_callback=false)

userData Issues userDataStream token and keepalive, subscribes to userData WebSocket
- **miniTicker** (Callable \$callback)

miniTicker Get miniTicker for all symbols

Public Attributes

- **\$httpDebug** = false
- **\$balances** = []
- **\$btc_value** = 0.00
- **\$btc_total** = 0.00

Protected Attributes

- **\$base** = "https://api.binance.com/api/"
- **\$wapi** = "https://api.binance.com/wapi/"
- **\$api_key**
- **\$api_secret**
- **\$depthCache** = []
- **\$depthQueue** = []
- **\$chartQueue** = []
- **\$charts** = []
- **\$info**
- **\$proxyConf** = null
- **\$transfered** = 0
- **\$requestCount** = 0

4.1.1 Detailed Description

Main Binance class.

Eg. Usage: require 'vendor/autoload.php'; \$api = new Binance\API();

Definition at line 23 of file php-binance-api.php.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 `Binance\API::__construct (string $api_key = "", string $api_secret = "", array $options = ["useServerTime"=, false], array $proxyConf = null)`

Constructor for the class, There are 4 ways to construct the class:

- You can use the config file in ~/jaggedsoft/php-binance-api.json and empty constructor
- `new Binance\API($api_key, $api_secret);`
- `new Binance\API($api_key, $api_secret, $options);`
- `new Binance\API($api_key, $api_secret, $options, $proxyConf);`

Parameters

<i>\$api_key</i>	string api key
<i>\$api_secret</i>	string api secret
<i>\$options</i>	array additional configuration options
<i>\$proxyConf</i>	array config

Returns

null

Definition at line 62 of file php-binance-api.php.

4.1.3 Member Function Documentation

4.1.3.1 `Binance\API::account ()`

account get all information about the api account

```
$account = $api->account();
```

Returns

array with error message or array of all the account information

Definition at line 562 of file php-binance-api.php.

4.1.3.2 `Binance\API::addToTransferred (int $int)`

addToTransferred add interger bytes to the total transferred also incrementes the api counter

```
$apiCount = $api->addToTransferred( $int );
```

Returns

null

Definition at line 1385 of file php-binance-api.php.

4.1.3.3 `Binance\API::aggTrades (string $symbol)`

aggTrades get Market History / Aggregate Trades

```
$trades = $api->aggTrades("BNBBTC");
```


Parameters

<i>\$symbol</i>	string the symbol to get the trade information for
-----------------	--

Returns

array with error message or array of market history

Definition at line 594 of file php-binance-api.php.

4.1.3.4 `Binance\API::balances ($priceData = false)`

balances get balances for the account assets

```
$balances = $api->balances($ticker);
```

Parameters

<i>\$priceData</i>	array of the symbols balances are required for
--------------------	--

Returns

array with error message or array of balances

Definition at line 632 of file php-binance-api.php.

4.1.3.5 `Binance\API::bookPrices ()`

bookPrices get all bid/asks prices

```
$ticker = $api->bookPrices();
```

Returns

array with error message or array of all the book prices

Definition at line 551 of file php-binance-api.php.

4.1.3.6 `Binance\API::buy (string $symbol, $quantity, $price, string $type = "LIMIT", array $flags = [])`

buy attempts to create a currency order each currency supports a number of order types, such as -LIMIT -MARKET -STOP_LOSS -STOP_LOSS_LIMIT -TAKE_PROFIT -TAKE_PROFIT_LIMIT -LIMIT_MAKER

You should check the

See Also

exchangeInfo for each currency to determine what types of orders can be placed against specific pairs

```
$quantity = 1; $price = 0.0005; $order = $api->buy("BNBBTC", $quantity, $price);
```

Parameters

<i>\$symbol</i>	string the currency symbol
<i>\$quantity</i>	string the quantity required

<i>\$price</i>	string price per unit you want to spend
<i>\$type</i>	string type of order
<i>\$flags</i>	array additional options for order type

Returns

array with error message or the order details

Definition at line 150 of file php-binance-api.php.

4.1.3.7 `Binance\API::buyTest (string $symbol, $quantity, $price, string $type = "LIMIT", array $flags = [])`

buyTest attempts to create a TEST currency order

See Also

buy()

Parameters

<i>\$symbol</i>	string the currency symbol
<i>\$quantity</i>	string the quantity required
<i>\$price</i>	string price per unit you want to spend
<i>\$type</i>	array config
<i>\$flags</i>	array config

Returns

array with error message or empty or the order details

Definition at line 171 of file php-binance-api.php.

4.1.3.8 `Binance\API::cancel (string $symbol, $orderid)`

cancel attempts to cancel a currency order

```
$orderid = "123456789"; $order = $api->cancel("BNBBTC", $orderid);
```

Parameters

<i>\$symbol</i>	string the currency symbol
<i>\$orderid</i>	string the orderid to cancel

Returns

array with error message or the order details

Definition at line 312 of file php-binance-api.php.

4.1.3.9 `Binance\API::candlesticks (string $symbol, string $interval = "5m", int $limit = null, $startTime = null, $endTime = null)`

candlesticks get the candles for the given intervals 1m,3m,5m,15m,30m,1h,2h,4h,6h,8h,12h,1d,3d,1w,1M

```
$candles = $api->candlesticks("BNBBTC", "5m");
```

Parameters

<i>\$symbol</i>	string to query
<i>\$interval</i>	string to request
<i>\$limit</i>	int limit the amount of candles
<i>\$startTime</i>	string request candle information starting from here
<i>\$endTime</i>	string request candle information ending here

Returns

array containing the response

Definition at line 919 of file php-binance-api.php.

4.1.3.10 **Binance\API::chart (*\$symbols*, string *\$interval* = "30m", Callable *\$callback*)**

chart Pulls /kline data and subscribes to WebSocket endpoint

```
$api->chart(["BNBBTC"], "15m", function($api, $symbol, $chart) { echo "{$symbol} chart update\n"; print_r($chart); });
```

Parameters

<i>\$symbols</i>	string required symbols
<i>\$interval</i>	string time interval
<i>\$callback</i>	callable closure

Returns

null

Definition at line 1662 of file php-binance-api.php.

4.1.3.11 **Binance\API::cumulative (array *\$depth*)**

cumulative Converts depth cache into a cumulative array

```
$cumulative = $api->cumulative($depth);
```

Parameters

<i>\$depth</i>	array cache array
----------------	-------------------

Returns

array cumulative depth cache

Definition at line 1212 of file php-binance-api.php.

4.1.3.12 **Binance\API::depositAddress (string *\$asset*)**

depositAddress get the deposit address for an asset

```
$depositAddress = $api->depositAddress("VEN");
```

Parameters

<i>\$asset</i>	string the currency such as BTC
----------------	---------------------------------

Returns

array with error message or array deposit address information

Definition at line 483 of file php-binance-api.php.

4.1.3.13 `BinanceAPI::depositHistory (string $asset = null)`

depositAddress get the deposit history for an asset

```
$depositHistory = $api->depositHistory();
```

```
$depositHistory = $api->depositHistory( "BTC" );
```

Parameters

<i>\$asset</i>	string empty or the currency such as BTC
----------------	--

Returns

array with error message or array deposit history information

Definition at line 502 of file php-binance-api.php.

4.1.3.14 `BinanceAPI::depth (string $symbol)`

depth get Market depth

```
$depth = $api->depth("ETHBTC");
```

Parameters

<i>\$symbol</i>	string the symbol to get the depth information for
-----------------	--

Returns

array with error message or array of market depth

Definition at line 609 of file php-binance-api.php.

4.1.3.15 `BinanceAPI::depthCache ($symbols, Callable $callback)`

depthCache Pulls /depth data and subscribes to WebSocket endpoint Maintains a local Depth Cache in sync via lastUpdateId.

See depth() and depthHandler()

```
$api->depthCache(["BNBBTC"], function($api, $symbol, $depth) { echo "{$symbol} depth cache update".PHP_EOL; //print_r($depth); // Print all depth data $limit = 11; // Show only the closest asks/bids $sorted = $api->sort-Depth($symbol, $limit); $bid = $api->first($sorted['bids']); $ask = $api->first($sorted['asks']); echo $api->display-Depth($sorted); echo "ask: {$ask}".PHP_EOL; echo "bid: {$bid}".PHP_EOL; });
```

Parameters

<i>\$symbol</i>	string optional array of symbols
<i>\$callback</i>	callable closure

Returns

null

Definition at line 1506 of file php-binance-api.php.

4.1.3.16 `Binance\API::displayDepth (array $array)`

displayDepth Formats nicely for console output

```
$outputString = $api->displayDepth($array);
```

Parameters

<i>\$array</i>	array
----------------	-------

Returns

string of the depth information

Definition at line 1302 of file php-binance-api.php.

4.1.3.17 `Binance\API::exchangeInfo ()`

exchangeInfo Gets the complete exchange info, including limits, currency options etc.

```
$info = $api->exchangeInfo();
```

Returns

array with error message or exchange info array

Definition at line 433 of file php-binance-api.php.

4.1.3.18 `Binance\API::first (array $array)`

first Gets first key of an array

```
$first = $api->first($array);
```

Parameters

<i>\$array</i>	array
----------------	-------

Returns

string key or null

Definition at line 1272 of file php-binance-api.php.

4.1.3.19 `Binance\API::getProxyUriString ()`

getProxyUriString get Uniform Resource Identifier string associated with proxy config

```
$balances = $api->getProxyUriString();
```

Returns

string uri

Definition at line 645 of file php-binance-api.php.

4.1.3.20 BinanceAPI::getRequestCount ()

getRequestCount gets the total number of API calls

```
$apiCount = $api->getRequestCount();
```

Returns

int get the total number of api calls

Definition at line 1373 of file php-binance-api.php.

4.1.3.21 BinanceAPI::getTransferred ()

getTransferred gets the total transferred in b,Kb,Mb,Gb

```
$transferred = $api->getTransferred();
```

Returns

string showing the total transferred

Definition at line 1354 of file php-binance-api.php.

4.1.3.22 BinanceAPI::highstock (array \$chart, bool \$include_volume = false)

highstock Converts Chart Data into array for highstock & kline charts

```
$highstock = $api->highstock($chart, $include_volume);
```

Parameters

<i>\$chart</i>	array
<i>\$include_volume</i>	bool for inclusion of volume

Returns

array highchart data

Definition at line 1247 of file php-binance-api.php.

4.1.3.23 BinanceAPI::history (string \$symbol, int \$limit = 500, int \$fromTraded = 1)

history Get the complete account trade history for all or a specific currency

```
$allHistory = $api->history(); $BNBHistory = $api->history("BNBBTC"); $limitBNBHistory = $api->history("BNBBTC",5); $limitBNBHistoryFromId = $api->history("BNBBTC",5,3);
```

Parameters

<i>\$symbol</i>	string the currency symbol
<i>\$limit</i>	int the amount of orders returned
<i>\$fromTradeId</i>	int return the orders from this order onwards

Returns

array with error message or array of orderDetails array

Definition at line 395 of file php-binance-api.php.

4.1.3.24 Binance\API::keepAlive ()

keepAlive Keep-alive function for userDataStream

```
$api->keepAlive();
```

Returns

null

Definition at line 1719 of file php-binance-api.php.

4.1.3.25 Binance\API::last (array \$array)

last Gets last key of an array

```
$last = $api->last($array);
```

Parameters

<i>\$array</i>	array
----------------	-------

Returns

string key or null

Definition at line 1287 of file php-binance-api.php.

4.1.3.26 Binance\API::marketBuy (string \$symbol, \$quantity, array \$flags = [])

marketBuy attempts to create a currency order at given market price

```
$quantity = 1; $order = $api->marketBuy("BNBBTC", $quantity);
```

Parameters

<i>\$symbol</i>	string the currency symbol
<i>\$quantity</i>	string the quantity required
<i>\$flags</i>	array additional options for order type

Returns

array with error message or the order details

Definition at line 244 of file php-binance-api.php.

4.1.3.27 `BinanceAPI::marketBuyTest (string $symbol, $quantity, array $flags = [])`

`marketBuyTest` attempts to create a TEST currency order at given market price

See Also

`marketBuy()`

Parameters

<i>\$symbol</i>	string the currency symbol
<i>\$quantity</i>	string the quantity required
<i>\$flags</i>	array additional options for order type

Returns

array with error message or the order details

Definition at line 261 of file `php-binance-api.php`.

4.1.3.28 `BinanceAPI::marketSell (string $symbol, $quantity, array $flags = [])`

`marketSell` attempts to create a currency order at given market price

```
$quantity = 1; $order = $api->marketSell("BNBBTC", $quantity);
```

Parameters

<i>\$symbol</i>	string the currency symbol
<i>\$quantity</i>	string the quantity required
<i>\$flags</i>	array additional options for order type

Returns

array with error message or the order details

Definition at line 279 of file `php-binance-api.php`.

4.1.3.29 `BinanceAPI::marketSellTest (string $symbol, $quantity, array $flags = [])`

`marketSellTest` attempts to create a TEST currency order at given market price

See Also

`marketSellTest()`

Parameters

<i>\$symbol</i>	string the currency symbol
<i>\$quantity</i>	string the quantity required
<i>\$flags</i>	array additional options for order type

Returns

array with error message or the order details

Definition at line 296 of file `php-binance-api.php`.

4.1.3.30 Binance\API::miniTicker (Callable *\$callback*)

miniTicker Get miniTicker for all symbols

```
$api->miniTicker(function($api, $ticker) { print_r($ticker); });
```

Parameters

<i>\$callback</i>	callable function closer that takes 2 arguments, \$pai and \$ticker data
-------------------	--

Returns

null

Definition at line 1806 of file php-binance-api.php.

4.1.3.31 `BinanceAPI::openOrders (string $symbol = null)`

openOrders attempts to get open orders for all currencies or a specific currency

```
$allOpenOrders = $api->openOrders(); $allBNBOrders = $api->openOrders( "BNBBTC" );
```

Parameters

<i>\$symbol</i>	string the currency symbol
-----------------	----------------------------

Returns

array with error message or the order details

Definition at line 348 of file php-binance-api.php.

4.1.3.32 `BinanceAPI::order (string $side, string $symbol, $quantity, $price, string $type = "LIMIT", array $flags = [], bool $test = false)`

order formats the orders before sending them to the curl wrapper function You can call this function directly or use the helper functions

See Also

```
buy()
sell()
marketBuy()
marketSell() $this->httpRequest( "https://api.binance.com/api/v1/ticker/24hr");
```

Parameters

<i>\$side</i>	string typically "BUY" or "SELL"
<i>\$symbol</i>	string to buy or sell
<i>\$quantity</i>	string in the order
<i>\$price</i>	string for the order
<i>\$type</i>	string is determined by the symbol bu typicall LIMIT, STOP_LOSS_LIMIT etc.
<i>\$flags</i>	array additional transaction options
<i>\$test</i>	bool whether to test or not, test only validates the query

Returns

array containing the response

Definition at line 836 of file php-binance-api.php.

4.1.3.33 `BinanceAPI::orders (string $symbol, int $limit = 500, int $fromOrderId = 1)`

orders attempts to get the orders for all or a specific currency

```
$allBNBOrders = $api->orders( "BNBBTC" );
```

Parameters

<i>\$symbol</i>	string the currency symbol
<i>\$limit</i>	int the amount of orders returned
<i>\$fromOrderId</i>	string return the orders from this order onwards

Returns

array with error message or array of orderDetails array

Definition at line 371 of file php-binance-api.php.

4.1.3.34 Binance\API::orderStatus (string \$symbol, \$orderid)

orderStatus attempts to get orders status

```
$orderid = "123456789"; $order = $api->orderStatus("BNBBTC", $orderid);
```

Parameters

<i>\$symbol</i>	string the currency symbol
<i>\$orderid</i>	string the orderid to cancel

Returns

array with error message or the order details

Definition at line 331 of file php-binance-api.php.

4.1.3.35 Binance\API::prevDay (string \$symbol)

prevDay get 24hr ticker price change statistics for symbols

```
$prevDay = $api->prevDay("BNBBTC");
```

Parameters

<i>\$symbol</i>	(optional) symbol to get the previous day change for
-----------------	--

Returns

array with error message or array of prevDay change

Definition at line 575 of file php-binance-api.php.

4.1.3.36 Binance\API::prices ()

prices get all the current prices

```
$ticker = $api->prices();
```

Returns

array with error message or array of all the currencies prices

Definition at line 540 of file php-binance-api.php.

4.1.3.37 `Binance\API::sell (string $symbol, $quantity, $price, string $type = "LIMIT", array $flags = [])`

`sell` attempts to create a currency order each currency supports a number of order types, such as -LIMIT -MARKET -STOP_LOSS -STOP_LOSS_LIMIT -TAKE_PROFIT -TAKE_PROFIT_LIMIT -LIMIT_MAKER

You should check the

See Also

`exchangeInfo` for each currency to determine what types of orders can be placed against specific pairs

```
$quantity = 1; $price = 0.0005; $order = $api->sell("BNBBTC", $quantity, $price);
```

Parameters

<i>\$symbol</i>	string the currency symbol
<i>\$quantity</i>	string the quantity required
<i>\$price</i>	string price per unit you want to spend
<i>\$type</i>	string type of order
<i>\$flags</i>	array additional options for order type

Returns

array with error message or the order details

Definition at line 205 of file `php-binance-api.php`.

4.1.3.38 `Binance\API::sellTest (string $symbol, $quantity, $price, string $type = "LIMIT", array $flags = [])`

`sellTest` attempts to create a TEST currency order

See Also

`sell()`

Parameters

<i>\$symbol</i>	string the currency symbol
<i>\$quantity</i>	string the quantity required
<i>\$price</i>	string price per unit you want to spend
<i>\$type</i>	array config
<i>\$flags</i>	array config

Returns

array with error message or empty or the order details

Definition at line 226 of file `php-binance-api.php`.

4.1.3.39 `Binance\API::setProxy (array $proxyconf)`

`setProxy` set proxy config by passing in an array of the proxy configuration

```
$proxyConf = [ 'proto' => 'tcp', 'address' => '192.168.1.1', 'port' => '8080', 'user' => 'dude', 'pass' => 'd00d' ];
$api->setProxy( $proxyconf );
```

Returns

null

Definition at line 693 of file `php-binance-api.php`.

4.1.3.40 `Binance\API::sortDepth (string $symbol, int $limit = 11)`

sortDepth Sorts depth data for display & getting highest bid and lowest ask

```
$sorted = $api->sortDepth($symbol, $limit);
```

Parameters

<i>\$symbol</i>	string to sort
<i>\$limit</i>	int depth

Returns

null

Definition at line 1472 of file php-binance-api.php.

4.1.3.41 `Binance\API::ticker ($symbol, Callable $callback)`

ticker pulls 24h price change statistics via WebSocket

```
$api->ticker(false, function($api, $symbol, $ticker) { print_r($ticker); });
```

Parameters

<i>\$symbol</i>	string optional symbol or false
<i>\$callback</i>	callable closure

Returns

null

Definition at line 1622 of file php-binance-api.php.

4.1.3.42 `Binance\API::time ()`

time Gets the server time

```
$time = $api->time();
```

Returns

array with error message or array with server time key

Definition at line 422 of file php-binance-api.php.

4.1.3.43 `Binance\API::trades ($symbols, Callable $callback)`

trades Trades WebSocket Endpoint

```
$api->trades(["BNBBTC"], function($api, $symbol, $trades) { echo "{$symbol} trades update".PHP_EOL; print_r($trades); });
```

Parameters

<i>\$symbol</i>	string optional symbol
-----------------	------------------------

<i>\$callback</i>	callable closure
-------------------	------------------

Returns

null

Definition at line 1568 of file php-binance-api.php.

4.1.3.44 `BinanceAPI::userData (& $balance_callback, & $execution_callback = false)`

userData Issues userDataStream token and keepalive, subscribes to userData WebSocket

```
$balance_update = function($api, $balances) { print_r($balances); echo "Balance update".PHP_EOL; };

$order_update = function($api, $report) { echo "Order update".PHP_EOL; print_r($report); $price = $report['price'];
$quantity = $report['quantity']; $symbol = $report['symbol']; $side = $report['side']; $orderType = $report['order-
Type']; $orderId = $report['orderId']; $orderStatus = $report['orderStatus']; $executionType = $report['orderStatus'];
if( $executionType == "NEW" ) { if( $executionType == "REJECTED" ) { echo "Order Failed! Reason: {$report['reject-
Reason']}'".PHP_EOL; } echo "{$symbol} {$side} {$orderType} ORDER #{$orderId} ({$orderStatus})".PHP_EOL;
echo "..price: {$price}, quantity: {$quantity}".PHP_EOL; return; }

//NEW, CANCELED, REPLACED, REJECTED, TRADE, EXPIRED echo "{$symbol} {$side} {$executionType}
{$orderType} ORDER #{$orderId}".PHP_EOL; }; $api->userData($balance_update, $order_update);
```

Parameters

<i>\$balance_ - callback</i>	callable function
<i>\$execution_ - callback</i>	callable function

Returns

null

Definition at line 1767 of file php-binance-api.php.

4.1.3.45 `BinanceAPI::useServerTime ()`

useServerTime adds the 'useServerTime'=>true to the API request to avoid time errors

```
$api->useServerTime();
```

Returns

null

Definition at line 410 of file php-binance-api.php.

4.1.3.46 `BinanceAPI::withdraw (string $asset, string $address, $amount, $addressTag = null)`

withdraw requests a asset be withdrawn from binance to another wallet

```
$asset = "BTC"; $address = "1C5gqLRs96Xq4V2ZZAR1347yUCpHie7sa"; $amount = 0.2; $response = $api-
>withdraw($asset, $address, $amount);
```

```
$address = "44tLjmXrQNrWJ5NBsEj2R77ZBEgDa3fEe9GLpSf2FRmhexPvfYDUAB7EXX1Hdb3aMQ9FLqdJ56ya-
AhiXoRsceGJCRS3Jxkn"; $addressTag = "0e5e38a01058dbf64e53a4333a5acf98e0d5feb8e523d32e3186c664a9c762c1";
$amount = 0.1; $response = $api->withdraw($asset, $address, $amount, $addressTag);
```

Parameters

<i>\$asset</i>	string the currency such as BTC
<i>\$address</i>	string the addressed to whihc the asset should be deposited
<i>\$amount</i>	double the amount of the asset to transfer
<i>\$addressTag</i>	string adtional transactionid required by some assets

Returns

array with error message or array transaction

Definition at line 460 of file php-binance-api.php.

4.1.3.47 `Binance\API::withdrawHistory (string $asset = null)`

withdrawHistory get the withdrawal history for an asset

```
$withdrawHistory = $api->withdrawHistory();
```

```
$withdrawHistory = $api->withdrawHistory( "BTC" );
```

Parameters

<i>\$asset</i>	string empty or the currency such as BTC
----------------	--

Returns

array with error message or array deposit history information

Definition at line 523 of file php-binance-api.php.

4.1.4 Member Data Documentation

4.1.4.1 `Binance\API::$api_key` [protected]

Definition at line 26 of file php-binance-api.php.

4.1.4.2 `Binance\API::$api_secret` [protected]

Definition at line 27 of file php-binance-api.php.

4.1.4.3 `Binance\API::$balances = []`

Definition at line 39 of file php-binance-api.php.

4.1.4.4 `Binance\API::$base = "https://api.binance.com/api/"` [protected]

Definition at line 24 of file php-binance-api.php.

4.1.4.5 `Binance\API::$btc_total = 0.00`

Definition at line 41 of file php-binance-api.php.

4.1.4.6 `Binance\API::$btc_value = 0.00`

Definition at line 40 of file php-binance-api.php.

4.1.4.7 `Binance\API::$chartQueue = []` [protected]

Definition at line 30 of file php-binance-api.php.

4.1.4.8 `Binance\API::$charts = []` [protected]

Definition at line 31 of file php-binance-api.php.

4.1.4.9 `Binance\API::$depthCache = []` [protected]

Definition at line 28 of file php-binance-api.php.

4.1.4.10 `Binance\API::$depthQueue = []` [protected]

Definition at line 29 of file php-binance-api.php.

4.1.4.11 `Binance\API::$httpDebug = false`

Definition at line 38 of file php-binance-api.php.

4.1.4.12 `Binance\API::$info` [protected]

Initial value:

```
= [
    "timeOffset" => 0
]
```

Definition at line 32 of file php-binance-api.php.

4.1.4.13 `Binance\API::$proxyConf = null` [protected]

Definition at line 35 of file php-binance-api.php.

4.1.4.14 `Binance\API::$requestCount = 0` [protected]

Definition at line 37 of file php-binance-api.php.

4.1.4.15 `Binance\API::$transferred = 0` [protected]

Definition at line 36 of file php-binance-api.php.

4.1.4.16 `Binance\API::$wapi = "https://api.binance.com/wapi/"` [protected]

Definition at line 25 of file php-binance-api.php.

Index

\$api_key
 Binance::API, 25
\$api_secret
 Binance::API, 25
\$balances
 Binance::API, 25
\$base
 Binance::API, 25
\$btc_total
 Binance::API, 25
\$btc_value
 Binance::API, 25
\$chartQueue
 Binance::API, 25
\$charts
 Binance::API, 26
\$depthCache
 Binance::API, 26
\$depthQueue
 Binance::API, 26
\$httpDebug
 Binance::API, 26
\$info
 Binance::API, 26
\$proxyConf
 Binance::API, 26
\$requestCount
 Binance::API, 26
\$transferred
 Binance::API, 26
\$wapi
 Binance::API, 26
__construct
 Binance::API, 10

account
 Binance::API, 10
addToTransferred
 Binance::API, 10
aggTrades
 Binance::API, 10

balances
 Binance::API, 11
Binance, 5
Binance::API
 \$api_key, 25
 \$api_secret, 25
 \$balances, 25
 \$base, 25
 \$btc_total, 25
 \$btc_value, 25
 \$chartQueue, 25
 \$charts, 26
 \$depthCache, 26
 \$depthQueue, 26
 \$httpDebug, 26
 \$info, 26
 \$proxyConf, 26
 \$requestCount, 26
 \$transferred, 26
 \$wapi, 26
 __construct, 10
 account, 10
 addToTransferred, 10
 aggTrades, 10
 balances, 11
 bookPrices, 11
 buy, 11
 buyTest, 12
 cancel, 12
 candlesticks, 12
 chart, 13
 cumulative, 13
 depositAddress, 13
 depositHistory, 14
 depth, 14
 depthCache, 14
 displayDepth, 15
 exchangeInfo, 15
 first, 15
 getProxyUriString, 15
 getRequestCount, 16
 getTransferred, 16
 highstock, 16
 history, 16
 keepAlive, 17
 last, 17
 marketBuy, 17
 marketBuyTest, 17
 marketSell, 18
 marketSellTest, 18
 miniTicker, 18
 openOrders, 20
 order, 20
 orderStatus, 21
 orders, 20
 prevDay, 21
 prices, 21

- sell, [21](#)
- sellTest, [22](#)
- setProxy, [22](#)
- sortDepth, [22](#)
- ticker, [23](#)
- time, [23](#)
- trades, [23](#)
- useServerTime, [24](#)
- userData, [24](#)
- withdraw, [24](#)
- withdrawHistory, [25](#)
- Binance\API, [7](#)
- bookPrices
 - Binance::API, [11](#)
- buy
 - Binance::API, [11](#)
- buyTest
 - Binance::API, [12](#)
- cancel
 - Binance::API, [12](#)
- candlesticks
 - Binance::API, [12](#)
- chart
 - Binance::API, [13](#)
- cumulative
 - Binance::API, [13](#)
- depositAddress
 - Binance::API, [13](#)
- depositHistory
 - Binance::API, [14](#)
- depth
 - Binance::API, [14](#)
- depthCache
 - Binance::API, [14](#)
- displayDepth
 - Binance::API, [15](#)
- exchangeInfo
 - Binance::API, [15](#)
- first
 - Binance::API, [15](#)
- getProxyUriString
 - Binance::API, [15](#)
- getRequestCount
 - Binance::API, [16](#)
- getTransferred
 - Binance::API, [16](#)
- highstock
 - Binance::API, [16](#)
- history
 - Binance::API, [16](#)
- keepAlive
 - Binance::API, [17](#)
- last
 - Binance::API, [17](#)
- marketBuy
 - Binance::API, [17](#)
- marketBuyTest
 - Binance::API, [17](#)
- marketSell
 - Binance::API, [18](#)
- marketSellTest
 - Binance::API, [18](#)
- miniTicker
 - Binance::API, [18](#)
- openOrders
 - Binance::API, [20](#)
- order
 - Binance::API, [20](#)
- orderStatus
 - Binance::API, [21](#)
- orders
 - Binance::API, [20](#)
- prevDay
 - Binance::API, [21](#)
- prices
 - Binance::API, [21](#)
- sell
 - Binance::API, [21](#)
- sellTest
 - Binance::API, [22](#)
- setProxy
 - Binance::API, [22](#)
- sortDepth
 - Binance::API, [22](#)
- ticker
 - Binance::API, [23](#)
- time
 - Binance::API, [23](#)
- trades
 - Binance::API, [23](#)
- useServerTime
 - Binance::API, [24](#)
- userData
 - Binance::API, [24](#)
- withdraw
 - Binance::API, [24](#)
- withdrawHistory
 - Binance::API, [25](#)