

# PHP Binance Api

Generated by Doxygen 1.8.6

Tue Apr 3 2018 16:40:04



# Contents

<b>1</b>	<b>Namespace Index</b>	<b>1</b>
1.1	Namespace List . . . . .	1
<b>2</b>	<b>Class Index</b>	<b>3</b>
2.1	Class List . . . . .	3
<b>3</b>	<b>Namespace Documentation</b>	<b>5</b>
3.1	Binance Namespace Reference . . . . .	5
<b>4</b>	<b>Class Documentation</b>	<b>7</b>
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 . . . . .	11
4.1.3.4	balances . . . . .	12
4.1.3.5	bookPrices . . . . .	12
4.1.3.6	buy . . . . .	12
4.1.3.7	buyTest . . . . .	13
4.1.3.8	cancel . . . . .	13
4.1.3.9	candlesticks . . . . .	14
4.1.3.10	chart . . . . .	15
4.1.3.11	cumulative . . . . .	15
4.1.3.12	depositAddress . . . . .	15
4.1.3.13	depositHistory . . . . .	16
4.1.3.14	depth . . . . .	16
4.1.3.15	depthCache . . . . .	16
4.1.3.16	displayDepth . . . . .	17
4.1.3.17	exchangeInfo . . . . .	17

4.1.3.18	first	17
4.1.3.19	getProxyUriString	18
4.1.3.20	getRequestCount	18
4.1.3.21	getTransferred	18
4.1.3.22	highstock	18
4.1.3.23	history	19
4.1.3.24	keepAlive	19
4.1.3.25	last	19
4.1.3.26	marketBuy	19
4.1.3.27	marketBuyTest	20
4.1.3.28	marketSell	20
4.1.3.29	marketSellTest	20
4.1.3.30	miniTicker	21
4.1.3.31	openOrders	21
4.1.3.32	order	21
4.1.3.33	orders	22
4.1.3.34	orderStatus	22
4.1.3.35	prevDay	23
4.1.3.36	prices	23
4.1.3.37	sell	23
4.1.3.38	sellTest	24
4.1.3.39	setProxy	24
4.1.3.40	sortDepth	24
4.1.3.41	terminate	25
4.1.3.42	ticker	25
4.1.3.43	time	25
4.1.3.44	trades	26
4.1.3.45	userData	27
4.1.3.46	useServerTime	27
4.1.3.47	withdraw	28
4.1.3.48	withdrawHistory	28
4.1.4	Member Data Documentation	28
4.1.4.1	\$api_key	28
4.1.4.2	\$api_secret	28
4.1.4.3	\$balances	29
4.1.4.4	\$base	29
4.1.4.5	\$btc_total	29
4.1.4.6	\$btc_value	29
4.1.4.7	\$chartQueue	29
4.1.4.8	\$charts	29

4.1.4.9	\$depthCache . . . . .	29
4.1.4.10	\$depthQueue . . . . .	29
4.1.4.11	\$httpDebug . . . . .	29
4.1.4.12	\$info . . . . .	29
4.1.4.13	\$proxyConf . . . . .	29
4.1.4.14	\$requestCount . . . . .	30
4.1.4.15	\$stream . . . . .	30
4.1.4.16	\$subscriptions . . . . .	30
4.1.4.17	\$transferred . . . . .	30
4.1.4.18	\$wapi . . . . .	30

<b>Index</b>	<b>31</b>
--------------	-----------



# Chapter 1

## Namespace Index

### 1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">Binance</a> . . . . .	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=null)
  - 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 transferred 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 lastUpdateId.*
- **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*
- **terminate** (\$endpoint)  
*terminate Terminates websocket endpoints.*
- **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
- **\$subscriptions** = []
- **\$balances** = []
- **\$btc\_value** = 0.00
- **\$btc\_total** = 0.00

## Protected Attributes

- **\$base** = 'https://api.binance.com/api/'
- **\$wapi** = 'https://api.binance.com/wapi/'
- **\$stream** = 'wss://stream.binance.com:9443/ws/'
- **\$api\_key**
- **\$api\_secret**
- **\$depthCache** = []
- **\$depthQueue** = []
- **\$chartQueue** = []
- **\$charts** = []
- **\$info**
- **\$proxyConf** = null
- **\$transferred** = 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

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

##### Returns

null

Definition at line 60 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

##### Exceptions

<code>\Exception</code>	
-------------------------	--

Definition at line 525 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 1277 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

**Exceptions**

<i>\Exception</i>	
-------------------	--

Definition at line 557 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**

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

**Returns**

array with error message or array of balances

**Exceptions**

<i>\Exception</i>	
-------------------	--

Definition at line 595 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

**Exceptions**

<i>\Exception</i>	
-------------------	--

Definition at line 513 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 143 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>	string config
<i>\$flags</i>	array config

## Returns

array with error message or empty or the order details

Definition at line 159 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

## Exceptions

<i>\Exception</i>
-------------------

Definition at line 277 of file php-binance-api.php.

**4.1.3.9** `BinanceAPI::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

## Exceptions

<i>\Exception</i>	
-------------------	--

Definition at line 833 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

## Exceptions

<i>\Exception</i>	
-------------------	--

Definition at line 1564 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 1107 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

**Exceptions**

<i>\Exception</i>	
-------------------	--

Definition at line 443 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

**Exceptions**

<i>\Exception</i>	
-------------------	--

Definition at line 462 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

**Exceptions**

<i>\Exception</i>	
-------------------	--

Definition at line 572 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 = $api->sortDepth($symbol, $limit); $bid = $api->first($sorted['bids']); $ask = $api->first($sorted['asks']); echo $api->displayDepth($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 1386 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 1196 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

**Exceptions**

<i>\Exception</i>
-------------------

Definition at line 396 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 1166 of file php-binance-api.php.

**4.1.3.19 BinanceAPI::getProxyUriString ( )**

getProxyUriString get Uniform Resource Identifier string associated with proxy config

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

**Returns**

string uri

Definition at line 608 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 1265 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 1246 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 1141 of file php-binance-api.php.



**4.1.3.23** `Binance\API::history ( string $symbol, int $limit = 500, int $fromTradeId = 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

**Exceptions**

<i>\Exception</i>	
-------------------	--

Definition at line 355 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 1639 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 1181 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 219 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 233 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 248 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 262 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, \$api and \$ticker data
-------------------	--

## Returns

null

Definition at line 1735 of file php-binance-api.php.

4.1.3.31 `Binance\API::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

## Exceptions

<i>\Exception</i>	
-------------------	--

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

4.1.3.32 `Binance\API::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

## Exceptions

<i>\Exception</i>	
-------------------	--

Definition at line 774 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

## Exceptions

<i>\Exception</i>	
-------------------	--

Definition at line 333 of file php-binance-api.php.

#### 4.1.3.34 BinanceAPI::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

## Exceptions

<i>\Exception</i>	
-------------------	--

Definition at line 295 of file php-binance-api.php.

4.1.3.35 **Binance\API::prevDay ( string \$symbol = null )**

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

## Exceptions

<i>\Exception</i>	
-------------------	--

Definition at line 538 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

## Exceptions

<i>\Exception</i>	
-------------------	--

Definition at line 501 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 188 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 204 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 656 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 1354 of file php-binance-api.php.

**4.1.3.41** `Binance\API::terminate ( $endpoint )`

terminate Terminates websocket endpoints.

View endpoints first: `print_r($api->subscriptions)`

```
$api->terminate('ethbtc_kline@5m');
```

## Returns

null

Definition at line 1627 of file php-binance-api.php.

**4.1.3.42** `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 1516 of file php-binance-api.php.

**4.1.3.43** `Binance\API::time ( )`

time Gets the server time

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

## Returns

array with error message or array with server time key

## Exceptions

<i>\Exception</i>
-------------------

Definition at line 384 of file php-binance-api.php.

#### 4.1.3.44 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>\$symbols</i>	
<i>\$callback</i>	callable closure

## Returns

null

Definition at line 1455 of file php-binance-api.php.

#### 4.1.3.45 Binance\API::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
bool	<i>\$execution_ - callback</i>	callable function

## Returns

null

## Exceptions

<i>\Exception</i>
-------------------

Definition at line 1686 of file php-binance-api.php.

#### 4.1.3.46 Binance\API::useServerTime ( )

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

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

## Returns

null

## Exceptions

<i>\Exception</i>
-------------------

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

#### 4.1.3.47 `Binance\API::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 = "44tLjmXrQNrWJ5NBsEj2R77ZBEgDa3fEe9GLpSf2FRmhhexPvfYDUAB7EXX1Hdb3aMQ9FLqdJ56ya-AhiXoRsceGJCRS3Jxkn"; $addressTag = "0e5e38a01058dbf64e53a4333a5acf98e0d5feb8e523d32e3186c664a9c762c1"; $amount = 0.1; $response = $api->withdraw($asset, $address, $amount, $addressTag);
```

##### Parameters

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

##### Returns

array with error message or array transaction

##### Exceptions

<code>\Exception</code>
-------------------------

Definition at line 420 of file php-binance-api.php.

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

withdrawHistory get the withdrawal history for an asset

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

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

##### Parameters

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

##### Returns

array with error message or array deposit history information

##### Exceptions

<code>\Exception</code>
-------------------------

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

### 4.1.4 Member Data Documentation

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

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

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

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

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

Definition at line 41 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 43 of file `php-binance-api.php`.

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

Definition at line 42 of file `php-binance-api.php`.

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

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

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

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

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

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

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

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

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

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

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

**Initial value:**

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

Definition at line 33 of file `php-binance-api.php`.

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

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

**4.1.4.14** `BinanceAPI::$requestCount = 0` [protected]

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

**4.1.4.15** `BinanceAPI::$stream = 'wss://stream.binance.com:9443/ws/'` [protected]

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

**4.1.4.16** `BinanceAPI::$subscriptions = []`

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

**4.1.4.17** `BinanceAPI::$transferred = 0` [protected]

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

**4.1.4.18** `BinanceAPI::$wapi = 'https://api.binance.com/wapi/'` [protected]

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

# Index

\$api\_key  
    Binance::API, [28](#)  
\$api\_secret  
    Binance::API, [28](#)  
\$balances  
    Binance::API, [28](#)  
\$base  
    Binance::API, [29](#)  
\$btc\_total  
    Binance::API, [29](#)  
\$btc\_value  
    Binance::API, [29](#)  
\$chartQueue  
    Binance::API, [29](#)  
\$charts  
    Binance::API, [29](#)  
\$depthCache  
    Binance::API, [29](#)  
\$depthQueue  
    Binance::API, [29](#)  
\$httpDebug  
    Binance::API, [29](#)  
\$info  
    Binance::API, [29](#)  
\$proxyConf  
    Binance::API, [29](#)  
\$requestCount  
    Binance::API, [29](#)  
\$stream  
    Binance::API, [30](#)  
\$subscriptions  
    Binance::API, [30](#)  
\$transferred  
    Binance::API, [30](#)  
\$wapi  
    Binance::API, [30](#)  
\_\_construct  
    Binance::API, [10](#)  
  
account  
    Binance::API, [10](#)  
addToTransferred  
    Binance::API, [10](#)  
aggTrades  
    Binance::API, [10](#)  
  
balances  
    Binance::API, [12](#)  
Binance, [5](#)  
Binance::API  
    \$api\_key, [28](#)  
    \$api\_secret, [28](#)  
    \$balances, [28](#)  
    \$base, [29](#)  
    \$btc\_total, [29](#)  
    \$btc\_value, [29](#)  
    \$chartQueue, [29](#)  
    \$charts, [29](#)  
    \$depthCache, [29](#)  
    \$depthQueue, [29](#)  
    \$httpDebug, [29](#)  
    \$info, [29](#)  
    \$proxyConf, [29](#)  
    \$requestCount, [29](#)  
    \$stream, [30](#)  
    \$subscriptions, [30](#)  
    \$transferred, [30](#)  
    \$wapi, [30](#)  
    \_\_construct, [10](#)  
    account, [10](#)  
    addToTransferred, [10](#)  
    aggTrades, [10](#)  
    balances, [12](#)  
    bookPrices, [12](#)  
    buy, [12](#)  
    buyTest, [13](#)  
    cancel, [13](#)  
    candlesticks, [13](#)  
    chart, [15](#)  
    cumulative, [15](#)  
    depositAddress, [15](#)  
    depositHistory, [16](#)  
    depth, [16](#)  
    depthCache, [16](#)  
    displayDepth, [17](#)  
    exchangeInfo, [17](#)  
    first, [17](#)  
    getProxyUriString, [18](#)  
    getRequestCount, [18](#)  
    getTransferred, [18](#)  
    highstock, [18](#)  
    history, [18](#)  
    keepAlive, [19](#)  
    last, [19](#)  
    marketBuy, [19](#)  
    marketBuyTest, [20](#)  
    marketSell, [20](#)  
    marketSellTest, [20](#)  
    miniTicker, [21](#)

- openOrders, [21](#)
- order, [21](#)
- orderStatus, [22](#)
- orders, [22](#)
- prevDay, [23](#)
- prices, [23](#)
- sell, [23](#)
- sellTest, [24](#)
- setProxy, [24](#)
- sortDepth, [24](#)
- terminate, [25](#)
- ticker, [25](#)
- time, [25](#)
- trades, [25](#)
- useServerTime, [27](#)
- userData, [27](#)
- withdraw, [27](#)
- withdrawHistory, [28](#)
- Binance\API, [7](#)
- bookPrices
  - Binance::API, [12](#)
- buy
  - Binance::API, [12](#)
- buyTest
  - Binance::API, [13](#)
- cancel
  - Binance::API, [13](#)
- candlesticks
  - Binance::API, [13](#)
- chart
  - Binance::API, [15](#)
- cumulative
  - Binance::API, [15](#)
- depositAddress
  - Binance::API, [15](#)
- depositHistory
  - Binance::API, [16](#)
- depth
  - Binance::API, [16](#)
- depthCache
  - Binance::API, [16](#)
- displayDepth
  - Binance::API, [17](#)
- exchangeInfo
  - Binance::API, [17](#)
- first
  - Binance::API, [17](#)
- getProxyUriString
  - Binance::API, [18](#)
- getRequestCount
  - Binance::API, [18](#)
- getTransferred
  - Binance::API, [18](#)
- highstock
  - Binance::API, [18](#)
- history
  - Binance::API, [18](#)
- keepAlive
  - Binance::API, [19](#)
- last
  - Binance::API, [19](#)
- marketBuy
  - Binance::API, [19](#)
- marketBuyTest
  - Binance::API, [20](#)
- marketSell
  - Binance::API, [20](#)
- marketSellTest
  - Binance::API, [20](#)
- miniTicker
  - Binance::API, [21](#)
- openOrders
  - Binance::API, [21](#)
- order
  - Binance::API, [21](#)
- orderStatus
  - Binance::API, [22](#)
- orders
  - Binance::API, [22](#)
- prevDay
  - Binance::API, [23](#)
- prices
  - Binance::API, [23](#)
- sell
  - Binance::API, [23](#)
- sellTest
  - Binance::API, [24](#)
- setProxy
  - Binance::API, [24](#)
- sortDepth
  - Binance::API, [24](#)
- terminate
  - Binance::API, [25](#)
- ticker
  - Binance::API, [25](#)
- time
  - Binance::API, [25](#)
- trades
  - Binance::API, [25](#)
- useServerTime
  - Binance::API, [27](#)
- userData
  - Binance::API, [27](#)
- withdraw
  - Binance::API, [27](#)
- withdrawHistory
  - Binance::API, [28](#)