

# PHP Binance Api

Generated by Doxygen 1.8.6

Wed Mar 28 2018 19:07:13



# 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	aggTrades . . . . .	10
4.1.3.3	balances . . . . .	10
4.1.3.4	bookPrices . . . . .	11
4.1.3.5	buy . . . . .	11
4.1.3.6	buyTest . . . . .	11
4.1.3.7	cancel . . . . .	12
4.1.3.8	candlesticks . . . . .	12
4.1.3.9	chart . . . . .	12
4.1.3.10	cumulative . . . . .	12
4.1.3.11	depositAddress . . . . .	13
4.1.3.12	depositHistory . . . . .	13
4.1.3.13	depth . . . . .	13
4.1.3.14	depthCache . . . . .	14
4.1.3.15	displayDepth . . . . .	14
4.1.3.16	exchangeInfo . . . . .	14
4.1.3.17	first . . . . .	14

4.1.3.18	getProxyUriString	14
4.1.3.19	getRequestCount	15
4.1.3.20	getTransferred	15
4.1.3.21	highstock	15
4.1.3.22	history	15
4.1.3.23	keepAlive	15
4.1.3.24	last	15
4.1.3.25	marketBuy	16
4.1.3.26	marketBuyTest	16
4.1.3.27	marketSell	16
4.1.3.28	marketSellTest	17
4.1.3.29	miniTicker	17
4.1.3.30	openOrders	17
4.1.3.31	order	17
4.1.3.32	orders	18
4.1.3.33	orderStatus	18
4.1.3.34	prevDay	19
4.1.3.35	prices	19
4.1.3.36	sell	19
4.1.3.37	sellTest	19
4.1.3.38	setProxy	20
4.1.3.39	sortDepth	20
4.1.3.40	ticker	20
4.1.3.41	time	20
4.1.3.42	trades	20
4.1.3.43	userData	20
4.1.3.44	useServerTime	21
4.1.3.45	withdraw	21
4.1.3.46	withdrawHistory	21
4.1.4	Member Data Documentation	21
4.1.4.1	\$api_key	21
4.1.4.2	\$api_secret	22
4.1.4.3	\$balances	22
4.1.4.4	\$base	22
4.1.4.5	\$btc_total	22
4.1.4.6	\$btc_value	22
4.1.4.7	\$chartQueue	22
4.1.4.8	\$charts	22
4.1.4.9	\$depthCache	22
4.1.4.10	\$depthQueue	22

---

4.1.4.11	<a href="#">\$httpDebug</a>	23
4.1.4.12	<a href="#">\$info</a>	23
4.1.4.13	<a href="#">\$proxyConf</a>	23
4.1.4.14	<a href="#">\$requestCount</a>	23
4.1.4.15	<a href="#">\$transferred</a>	23
4.1.4.16	<a href="#">\$wapi</a>	23
<b>Index</b>		<b>24</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** (\$api\_key= "", \$api\_secret= "", \$options=["useServerTime"=>false], \$proxyConf=null)  
*Constructor for the class, There are 4 ways to construct the class:*
- **buy** (\$symbol, \$quantity, \$price, \$type="LIMIT", \$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** (\$symbol, \$quantity, \$price, \$type="LIMIT", \$flags=[])  
*buyTest attempts to create a TEST currency order*
- **sell** (\$symbol, \$quantity, \$price, \$type="LIMIT", \$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** (\$symbol, \$quantity, \$price, \$type="LIMIT", \$flags=[])  
*sellTest attempts to create a TEST currency order*
- **marketBuy** (\$symbol, \$quantity, \$flags=[])  
*marketBuy attempts to create a currency order at given market price*
- **marketBuyTest** (\$symbol, \$quantity, \$flags=[])  
*marketBuyTest attempts to create a TEST currency order at given market price*
- **marketSell** (\$symbol, \$quantity, \$flags=[])  
*marketSell attempts to create a currency order at given market price*
- **marketSellTest** (\$symbol, \$quantity, \$flags=[])  
*marketSellTest attempts to create a TEST currency order at given market price*
- **cancel** (\$symbol, \$orderid)  
*cancel attempts to cancel a currency order*
- **orderStatus** (\$symbol, \$orderid)  
*orderStatus attempts to get orders status*
- **openOrders** (\$symbol=null)  
*openOrders attempts to get open orders for all currencies or a specific currency*
- **orders** (\$symbol, \$limit=500, \$fromOrderId=1)  
*orders attempts to get the orders for all or a specific currency*
- **history** (\$symbol, \$limit=500, \$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](#) (\$asset, \$address, \$amount, \$addressTag=false)  
*withdraw requests a asset be withdrawn from binance to another wallet*
- [depositAddress](#) (\$asset)  
*depositAddress get the deposit address for an asset*
- [depositHistory](#) (\$asset=false)  
*depositAddress get the deposit history for an asset*
- [withdrawHistory](#) (\$asset=false)  
*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](#) (\$symbol)  
*prevDay get 24hr ticker price change statistics for a symbol*
- [aggTrades](#) (\$symbol)  
*aggTrades get Market History / Aggregate Trades*
- [depth](#) (\$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](#) (\$proxyconf)  
*setProxy set proxy config by passing in an array of the proxy configuration*
- [order](#) (\$side, \$symbol, \$quantity, \$price, \$type="LIMIT", \$flags=[], \$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](#) (\$symbol, \$interval="5m", \$limit=null, \$startTime=null, \$endTime=null)  
*candlesticks get the cancles for the given intervals 1m,3m,5m,15m,30m,1h,2h,4h,6h,8h,12h,1d,3d,1w,1M*
- [cumulative](#) (\$depth)  
*cumulative Converts depth cache into a cumulative array*
- [highstock](#) (\$chart, \$include\_volume=false)  
*highstock Converts Chart Data into array for highstock & kline charts*
- [first](#) (\$array)  
*first Gets first key of an array*
- [last](#) (\$array)  
*last Gets last key of an array*
- [displayDepth](#) (\$array)  
*displayDepth Formats nicely for console output*
- [sortDepth](#) (\$symbol, \$limit=11)
- [depthCache](#) (\$symbols, \$callback)
- [trades](#) (\$symbols, \$callback)
- [ticker](#) (\$symbol, \$callback)

- `chart` (\$symbols, \$interval="30m", \$callback)
- `keepAlive` ()
- `userData` (&\$balance\_callback, &\$execution\_callback=false)
- `miniTicker` (\$callback)
- `getTransferred` ()
- `getRequestCount` ()

## Public Attributes

- `$httpDebug` = false  
*If you enable this, curl will output debugging information.*
- `$balances` = []  
*binance balances from the last run*
- `$btc_value` = 0.00  
*value of available assets*
- `$btc_total` = 0.00  
*value of available + onOrder assets*

## Protected Attributes

- `$base` = "https://api.binance.com/api/"  
*REST endpoint for the currency exchange.*
- `$wapi` = "https://api.binance.com/wapi/"  
*REST endpoint for the withdrawals.*
- `$api_key`  
*API key that you created in the binance website member area.*
- `$api_secret`  
*API secret that was given to you when you created the api key.*
- `$depthCache` = []  
*Websockets depth cache.*
- `$depthQueue` = []  
*Websockets depth queue.*
- `$chartQueue` = []  
*Websockets chart queue.*
- `$charts` = []  
*Websockets chart data.*
- `$info` = ["timeOffset">0]  
*Additional connection options.*
- `$proxyConf` = null  
*Used for story the proxy configuration.*
- `$transferred` = 0  
*This stores the amount of bytes transferred.*
- `$requestCount` = 0  
*This stores the amount of API requests.*

### 4.1.1 Detailed Description

Main Binance class.

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

Definition at line 20 of file `php-binance-api.php`.

## 4.1.2 Constructor & Destructor Documentation

### 4.1.2.1 `Binance\API::__construct ( $api_key = "", $api_secret = "", $options = ["useServerTime"=, false], $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

nothing

Definition at line 51 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 457 of file `php-binance-api.php`.

### 4.1.3.2 `Binance\API::aggTrades ( $symbol )`

aggTrades get Market History / Aggregate Trades

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

#### Parameters

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

#### Returns

array with error message or array of market history

Definition at line 481 of file `php-binance-api.php`.

### 4.1.3.3 `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 508 of file php-binance-api.php.

4.1.3.4 `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 446 of file php-binance-api.php.

4.1.3.5 `Binance\API::buy ( $symbol, $quantity, $price, $type = "LIMIT", $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>	the currency symbol
<i>\$quantity</i>	the quantity required
<i>\$price</i>	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 125 of file php-binance-api.php.

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

buyTest attempts to create a TEST currency order

## See Also

buy()

**Parameters**

<i>\$symbol</i>	the currency symbol
<i>\$quantity</i>	the quantity required
<i>\$price</i>	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 140 of file php-binance-api.php.

**4.1.3.7 Binance\API::cancel ( *\$symbol*, *\$orderid* )**

cancel attempts to cancel a currency order

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

**Parameters**

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

**Returns**

array with error message or the order details

Definition at line 254 of file php-binance-api.php.

**4.1.3.8 Binance\API::candlesticks ( *\$symbol*, *\$interval* = "5m", *\$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>	to query
<i>\$interval</i>	to request
<i>\$limit</i>	limit the amount of candles
<i>\$startTime</i>	request candle information starting from here
<i>\$endTime</i>	request candle information ending here

**Returns**

array containing the response

Definition at line 700 of file php-binance-api.php.

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

Definition at line 1197 of file php-binance-api.php.

**4.1.3.10 Binance\API::cumulative ( *\$depth* )**

cumulative Converts depth cache into a cumulative array

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

## Parameters

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

## Returns

array cumulative depth cache

Definition at line 944 of file php-binance-api.php.

**4.1.3.11** `Binance\API::depositAddress ( $asset )`

depositAddress get the deposit address for an asset

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

## Parameters

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

## Returns

array with error message or array deposit address information

Definition at line 387 of file php-binance-api.php.

**4.1.3.12** `Binance\API::depositHistory ( $asset = false )`

depositAddress get the deposit history for an asset

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

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

## Parameters

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

## Returns

array with error message or array deposit history information

Definition at line 402 of file php-binance-api.php.

**4.1.3.13** `Binance\API::depth ( $symbol )`

depth get Market depth

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

## Parameters

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

## Returns

array with error message or array of market depth

Definition at line 493 of file php-binance-api.php.

#### 4.1.3.14 **Binance\API::depthCache** ( *\$symbols*, *\$callback* )

Definition at line 1103 of file php-binance-api.php.

#### 4.1.3.15 **Binance\API::displayDepth** ( *\$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 1022 of file php-binance-api.php.

#### 4.1.3.16 **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 350 of file php-binance-api.php.

#### 4.1.3.17 **Binance\API::first** ( *\$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 992 of file php-binance-api.php.

#### 4.1.3.18 **Binance\API::getProxyUriString** ( )

getProxyUriString get Uniform Resource Identifier string associated with proxy config

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

##### Returns

string uri

Definition at line 519 of file php-binance-api.php.

**4.1.3.19** `Binance\API::getRequestCount ( )`

Definition at line 1307 of file php-binance-api.php.

**4.1.3.20** `Binance\API::getTransferred ( )`

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

**4.1.3.21** `Binance\API::highstock ( $chart, $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 968 of file php-binance-api.php.

**4.1.3.22** `Binance\API::history ( $symbol, $limit = 500, $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>	the currency symbol
<i>\$limit</i>	the amount of orders returned
<i>\$fromTradeId</i>	return the orders from this order onwards

**Returns**

array with error message or array of orderDetails array

Definition at line 316 of file php-binance-api.php.

**4.1.3.23** `Binance\API::keepAlive ( )`

Definition at line 1240 of file php-binance-api.php.

**4.1.3.24** `Binance\API::last ( $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 1007 of file php-binance-api.php.

**4.1.3.25 BinanceAPI::marketBuy ( *\$symbol*, *\$quantity*, *\$flags* = [ ] )**

marketBuy attempts to create a currency order at given market price

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

**Parameters**

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

**Returns**

array with error message or the order details

Definition at line 199 of file php-binance-api.php.

**4.1.3.26 BinanceAPI::marketBuyTest ( *\$symbol*, *\$quantity*, *\$flags* = [ ] )**

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

**See Also**

marketBuy()

**Parameters**

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

**Returns**

array with error message or the order details

Definition at line 212 of file php-binance-api.php.

**4.1.3.27 BinanceAPI::marketSell ( *\$symbol*, *\$quantity*, *\$flags* = [ ] )**

marketSell attempts to create a currency order at given market price

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

## Parameters

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

## Returns

array with error message or the order details

Definition at line 227 of file php-binance-api.php.

#### 4.1.3.28 Binance\API::marketSellTest ( *\$symbol*, *\$quantity*, *\$flags* = [] )

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

## See Also

marketSellTest()

## Parameters

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

## Returns

array with error message or the order details

Definition at line 240 of file php-binance-api.php.

#### 4.1.3.29 Binance\API::miniTicker ( *\$callback* )

Definition at line 1275 of file php-binance-api.php.

#### 4.1.3.30 Binance\API::openOrders ( *\$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>	the currency symbol
-----------------	---------------------

## Returns

array with error message or the order details

Definition at line 281 of file php-binance-api.php.

#### 4.1.3.31 Binance\API::order ( *\$side*, *\$symbol*, *\$quantity*, *\$price*, *\$type* = "LIMIT", *\$flags* = [], *\$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>	typically "BUY" or "SELL"
<i>\$symbol</i>	to buy or sell
<i>\$quantity</i>	in the order
<i>\$price</i>	for the order
<i>\$type</i>	is determined by the symbol bu typical LIMIT, STOP_LOSS_LIMIT etc.
<i>\$flags</i>	additional transaction options
<i>\$test</i>	whether to test or not, test only validates the query

**Returns**

array containing the response

Definition at line 662 of file php-binance-api.php.

**4.1.3.32 BinanceAPI::orders ( *\$symbol*, *\$limit* = 500, *\$fromOrderId* = 1 )**

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

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

**Parameters**

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

**Returns**

array with error message or array of orderDetails array

Definition at line 299 of file php-binance-api.php.

**4.1.3.33 BinanceAPI::orderStatus ( *\$symbol*, *\$orderid* )**

orderStatus attempts to get orders status

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

**Parameters**

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

**Returns**

array with error message or the order details

Definition at line 268 of file php-binance-api.php.



**4.1.3.34** `Binance\API::prevDay ( $symbol )`

prevDay get 24hr ticker price change statistics for a symbol

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

**Parameters**

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

**Returns**

array with error message or array of prevDay change

Definition at line 469 of file php-binance-api.php.

**4.1.3.35** `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 435 of file php-binance-api.php.

**4.1.3.36** `Binance\API::sell ( $symbol, $quantity, $price, $type = "LIMIT", $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>	the currency symbol
<i>\$quantity</i>	the quantity required
<i>\$price</i>	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 169 of file php-binance-api.php.

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

sellTest attempts to create a TEST currency order

**See Also**

sell()

**Parameters**

<i>\$symbol</i>	the currency symbol
<i>\$quantity</i>	the quantity required
<i>\$price</i>	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 184 of file php-binance-api.php.

**4.1.3.38 BinanceAPI::setProxy ( \$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**

nothing

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

**4.1.3.39 BinanceAPI::sortDepth ( \$symbol, \$limit = 11 )**

Definition at line 1094 of file php-binance-api.php.

**4.1.3.40 BinanceAPI::ticker ( \$symbol, \$callback )**

Definition at line 1174 of file php-binance-api.php.

**4.1.3.41 BinanceAPI::time ( )**

time Gets the server time

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

**Returns**

array with error message or array with server time key

Definition at line 339 of file php-binance-api.php.

**4.1.3.42 BinanceAPI::trades ( \$symbols, \$callback )**

Definition at line 1142 of file php-binance-api.php.

**4.1.3.43 BinanceAPI::userData ( & \$balance\_callback, & \$execution\_callback = false )**

Definition at line 1249 of file php-binance-api.php.

4.1.3.44 `Binance\API::useServerTime ( )`

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

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

**Returns**

nothing

Definition at line 327 of file `php-binance-api.php`.

4.1.3.45 `Binance\API::withdraw ( $asset, $address, $amount, $addressTag = false )`

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

**Returns**

array with error message or array transaction

Definition at line 373 of file `php-binance-api.php`.

4.1.3.46 `Binance\API::withdrawHistory ( $asset = false )`

`withdrawHistory` get the withdrawal history for an asset

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

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

**Parameters**

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

**Returns**

array with error message or array deposit history information

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

## 4.1.4 Member Data Documentation

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

API key that you created in the binance website member area.

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

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

API secret that was given to you when you created the api key.

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

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

binace balances from the last run

Definition at line 34 of file php-binance-api.php.

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

REST endpoint for the currency exchange.

Definition at line 21 of file php-binance-api.php.

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

value of available + onOrder assets

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

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

value of available assets

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

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

Websockets chart queue.

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

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

Websockets chart data.

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

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

Websockets depth cache.

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

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

Websockets depth queue.

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

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

If you enable this, curl will output debugging information.

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

#### 4.1.4.12 `Binance\API::$info = ["timeOffset"=>0]` `[protected]`

Additional connection options.

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

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

Used for store the proxy configuration.

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

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

This stores the amount of API requests.

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

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

This stores the amount of bytes transferred.

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

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

REST endpoint for the withdrawals.

Definition at line 22 of file `php-binance-api.php`.

# Index

\$api\_key  
    Binance::API, [21](#)  
\$api\_secret  
    Binance::API, [21](#)  
\$balances  
    Binance::API, [22](#)  
\$base  
    Binance::API, [22](#)  
\$btc\_total  
    Binance::API, [22](#)  
\$btc\_value  
    Binance::API, [22](#)  
\$chartQueue  
    Binance::API, [22](#)  
\$charts  
    Binance::API, [22](#)  
\$depthCache  
    Binance::API, [22](#)  
\$depthQueue  
    Binance::API, [22](#)  
\$httpDebug  
    Binance::API, [22](#)  
\$info  
    Binance::API, [23](#)  
\$proxyConf  
    Binance::API, [23](#)  
\$requestCount  
    Binance::API, [23](#)  
\$transferred  
    Binance::API, [23](#)  
\$wapi  
    Binance::API, [23](#)  
\_\_construct  
    Binance::API, [10](#)  
  
account  
    Binance::API, [10](#)  
aggTrades  
    Binance::API, [10](#)  
  
balances  
    Binance::API, [10](#)  
Binance, [5](#)  
Binance::API  
    \$api\_key, [21](#)  
    \$api\_secret, [21](#)  
    \$balances, [22](#)  
    \$base, [22](#)  
    \$btc\_total, [22](#)  
    \$btc\_value, [22](#)  
    \$chartQueue, [22](#)  
    \$charts, [22](#)  
    \$depthCache, [22](#)  
    \$depthQueue, [22](#)  
    \$httpDebug, [22](#)  
    \$info, [23](#)  
    \$proxyConf, [23](#)  
    \$requestCount, [23](#)  
    \$transferred, [23](#)  
    \$wapi, [23](#)  
    \_\_construct, [10](#)  
    account, [10](#)  
    aggTrades, [10](#)  
    balances, [10](#)  
    bookPrices, [11](#)  
    buy, [11](#)  
    buyTest, [11](#)  
    cancel, [12](#)  
    candlesticks, [12](#)  
    chart, [12](#)  
    cumulative, [12](#)  
    depositAddress, [13](#)  
    depositHistory, [13](#)  
    depth, [13](#)  
    depthCache, [13](#)  
    displayDepth, [14](#)  
    exchangeInfo, [14](#)  
    first, [14](#)  
    getProxyUriString, [14](#)  
    getRequestCount, [14](#)  
    getTransferred, [15](#)  
    highstock, [15](#)  
    history, [15](#)  
    keepAlive, [15](#)  
    last, [15](#)  
    marketBuy, [16](#)  
    marketBuyTest, [16](#)  
    marketSell, [16](#)  
    marketSellTest, [17](#)  
    miniTicker, [17](#)  
    openOrders, [17](#)  
    order, [17](#)  
    orderStatus, [18](#)  
    orders, [18](#)  
    prevDay, [18](#)  
    prices, [19](#)  
    sell, [19](#)  
    sellTest, [19](#)  
    setProxy, [20](#)

- sortDepth, [20](#)
- ticker, [20](#)
- time, [20](#)
- trades, [20](#)
- useServerTime, [20](#)
- userData, [20](#)
- withdraw, [21](#)
- withdrawHistory, [21](#)
- Binance\API, [7](#)
- bookPrices
  - Binance::API, [11](#)
- buy
  - Binance::API, [11](#)
- buyTest
  - Binance::API, [11](#)
- cancel
  - Binance::API, [12](#)
- candlesticks
  - Binance::API, [12](#)
- chart
  - Binance::API, [12](#)
- cumulative
  - Binance::API, [12](#)
- depositAddress
  - Binance::API, [13](#)
- depositHistory
  - Binance::API, [13](#)
- depth
  - Binance::API, [13](#)
- depthCache
  - Binance::API, [13](#)
- displayDepth
  - Binance::API, [14](#)
- exchangeInfo
  - Binance::API, [14](#)
- first
  - Binance::API, [14](#)
- getProxyUriString
  - Binance::API, [14](#)
- getRequestCount
  - Binance::API, [14](#)
- getTransferred
  - Binance::API, [15](#)
- highstock
  - Binance::API, [15](#)
- history
  - Binance::API, [15](#)
- keepAlive
  - Binance::API, [15](#)
- last
  - Binance::API, [15](#)
- marketBuy
  - Binance::API, [16](#)
- marketBuyTest
  - Binance::API, [16](#)
- marketSell
  - Binance::API, [16](#)
- marketSellTest
  - Binance::API, [17](#)
- miniTicker
  - Binance::API, [17](#)
- openOrders
  - Binance::API, [17](#)
- order
  - Binance::API, [17](#)
- orderStatus
  - Binance::API, [18](#)
- orders
  - Binance::API, [18](#)
- prevDay
  - Binance::API, [18](#)
- prices
  - Binance::API, [19](#)
- sell
  - Binance::API, [19](#)
- sellTest
  - Binance::API, [19](#)
- setProxy
  - Binance::API, [20](#)
- sortDepth
  - Binance::API, [20](#)
- ticker
  - Binance::API, [20](#)
- time
  - Binance::API, [20](#)
- trades
  - Binance::API, [20](#)
- useServerTime
  - Binance::API, [20](#)
- userData
  - Binance::API, [20](#)
- withdraw
  - Binance::API, [21](#)
- withdrawHistory
  - Binance::API, [21](#)