

## PHP Binance Api

Generated by Doxygen 1.8.6

Sun Apr 22 2018 10:06:18



# 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	__get . . . . .	10
4.1.3.2	__set . . . . .	10
4.1.3.3	account . . . . .	10
4.1.3.4	addToTransferred . . . . .	10
4.1.3.5	aggTrades . . . . .	11
4.1.3.6	balances . . . . .	11
4.1.3.7	bookPrices . . . . .	11
4.1.3.8	buy . . . . .	12
4.1.3.9	buyTest . . . . .	12
4.1.3.10	cancel . . . . .	12
4.1.3.11	candlesticks . . . . .	13
4.1.3.12	chart . . . . .	13
4.1.3.13	cumulative . . . . .	14
4.1.3.14	depositAddress . . . . .	14
4.1.3.15	depositHistory . . . . .	14
4.1.3.16	depth . . . . .	15
4.1.3.17	depthCache . . . . .	15

4.1.3.18	displayDepth . . . . .	15
4.1.3.19	exchangeInfo . . . . .	16
4.1.3.20	first . . . . .	16
4.1.3.21	getProxyUriString . . . . .	16
4.1.3.22	getRequestCount . . . . .	16
4.1.3.23	getTransferred . . . . .	17
4.1.3.24	highstock . . . . .	17
4.1.3.25	history . . . . .	17
4.1.3.26	keepAlive . . . . .	18
4.1.3.27	kline . . . . .	18
4.1.3.28	last . . . . .	18
4.1.3.29	marketBuy . . . . .	18
4.1.3.30	marketBuyTest . . . . .	19
4.1.3.31	marketSell . . . . .	19
4.1.3.32	marketSellTest . . . . .	19
4.1.3.33	miniTicker . . . . .	20
4.1.3.34	openOrders . . . . .	20
4.1.3.35	order . . . . .	20
4.1.3.36	orders . . . . .	21
4.1.3.37	orderStatus . . . . .	21
4.1.3.38	prevDay . . . . .	22
4.1.3.39	prices . . . . .	22
4.1.3.40	sell . . . . .	22
4.1.3.41	sellTest . . . . .	23
4.1.3.42	setProxy . . . . .	23
4.1.3.43	sortDepth . . . . .	23
4.1.3.44	terminate . . . . .	24
4.1.3.45	ticker . . . . .	24
4.1.3.46	time . . . . .	24
4.1.3.47	trades . . . . .	25
4.1.3.48	userData . . . . .	26
4.1.3.49	useServerTime . . . . .	26
4.1.3.50	withdraw . . . . .	27
4.1.3.51	withdrawFee . . . . .	27
4.1.3.52	withdrawHistory . . . . .	27
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	\$base . . . . .	28
4.1.4.4	\$chartQueue . . . . .	28

4.1.4.5	\$charts	28
4.1.4.6	\$curlOpts	28
4.1.4.7	\$depthCache	28
4.1.4.8	\$depthQueue	28
4.1.4.9	\$info	28
4.1.4.10	\$proxyConf	29
4.1.4.11	\$requestCount	29
4.1.4.12	\$stream	29
4.1.4.13	\$transferred	29
4.1.4.14	\$wapi	29

<b>Index</b>	<b>30</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:

<a href="#">Binance\API</a>	
Main Binance class . . . . .	<a href="#">7</a>



## 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](#) ()  
*Constructor for the class, send as many argument as you want.*
- [\\_\\_get](#) (string \$member)  
*magic get for private and protected members*
- [\\_\\_set](#) (string \$member, \$value)  
*magic set for private and protected members*
- [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, array \$params=[])  
*depositAddress get the deposit history for an asset*
- **withdrawHistory** (string \$asset=null, array \$params=[])  
*withdrawHistory get the withdrawal history for an asset*
- **withdrawFee** (string \$asset)  
*withdrawFee get the withdrawal fee 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 associated 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 candles 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 transferred in b,Kb,Mb,Gb*
- **getRequestCount** ()  
*getRequestCount gets the total number of API calls*
- **addToTransferred** (int \$int)  
*addToTransferred add integer 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 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*
- **kline** (\$symbols, string \$interval="30m", callable \$callback)  
*kline Subscribes to WebSocket endpoint for latest chart data only*
- **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*

## 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** = []
- **\$curlOpts** = []
- **\$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 29 of file php-binance-api.php.

## 4.1.2 Constructor & Destructor Documentation

### 4.1.2.1 Binance\API::\_\_construct ( )

Constructor for the class, send as many argument as you want.

#### Returns

null

Definition at line 60 of file php-binance-api.php.

## 4.1.3 Member Function Documentation

### 4.1.3.1 Binance\API::\_\_get ( string \$member )

magic get for private and protected members

#### Parameters

<i>\$file</i>	string the name of the property to return
---------------	---

#### Returns

null

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

### 4.1.3.2 Binance\API::\_\_set ( string \$member, \$value )

magic set for private and protected members

#### Parameters

<i>\$member</i>	string the name of the member property
<i>\$value</i>	the value of the member property

Definition at line 185 of file php-binance-api.php.

### 4.1.3.3 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

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

Definition at line 724 of file php-binance-api.php.

### 4.1.3.4 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 1555 of file php-binance-api.php.

**4.1.3.5 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 758 of file php-binance-api.php.

**4.1.3.6 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 799 of file php-binance-api.php.

**4.1.3.7 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 711 of file php-binance-api.php.

#### 4.1.3.8 `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 298 of file php-binance-api.php.

#### 4.1.3.9 `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 315 of file php-binance-api.php.

#### 4.1.3.10 `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 440 of file php-binance-api.php.

#### 4.1.3.11 `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

## Exceptions

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

Definition at line 1065 of file php-binance-api.php.

#### 4.1.3.12 `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 1874 of file php-binance-api.php.

**4.1.3.13 BinanceAPI::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 1373 of file php-binance-api.php.

**4.1.3.14 BinanceAPI::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 621 of file php-binance-api.php.

**4.1.3.15 BinanceAPI::depositHistory ( string \$asset = null, array \$params = [] )**

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
<i>\$params</i>	array optional startTime, endTime, status parameters

**Returns**

array with error message or array deposit history information

## Exceptions

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

Definition at line 642 of file php-binance-api.php.

4.1.3.16 **Binance\API::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 774 of file php-binance-api.php.

4.1.3.17 **Binance\API::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 1676 of file php-binance-api.php.

4.1.3.18 **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 1468 of file php-binance-api.php.

**4.1.3.19 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 572 of file php-binance-api.php.

**4.1.3.20 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 1436 of file php-binance-api.php.

**4.1.3.21 Binance\API::getProxyUriString ( )**

getProxyUriString get Uniform Resource Identifier string associated with proxy config

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

**Returns**

string uri

Definition at line 815 of file php-binance-api.php.

**4.1.3.22 Binance\API::getRequestCount ( )**

getRequestCount gets the total number of API calls

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

**Returns**

int get the total number of api calls

Definition at line 1542 of file php-binance-api.php.

**4.1.3.23 Binance\API::getTransferred ( )**

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

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

**Returns**

string showing the total transferred

Definition at line 1522 of file php-binance-api.php.

**4.1.3.24 Binance\API::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 1408 of file php-binance-api.php.

**4.1.3.25 Binance\API::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>\$fromTraded</i>	int (optional) return the orders from this order onwards. negative for all

**Returns**

array with error message or array of orderDetails array

**Exceptions**

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

Definition at line 522 of file php-binance-api.php.

**4.1.3.26** `BinanceAPI::keepAlive ( )`

keepAlive Keep-alive function for userDataStream

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

**Returns**

null

Definition at line 2015 of file php-binance-api.php.

**4.1.3.27** `BinanceAPI::kline ( $symbols, string $interval = "30m", callable $callback )`

kline Subscribes to WebSocket endpoint for latest chart data only

```
$api->kline(["BNBBTC"], "15m", function($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 1955 of file php-binance-api.php.

**4.1.3.28** `BinanceAPI::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 1452 of file php-binance-api.php.

**4.1.3.29** `BinanceAPI::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 378 of file php-binance-api.php.

**4.1.3.30** `Binance\API::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 393 of file php-binance-api.php.

**4.1.3.31** `Binance\API::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 409 of file php-binance-api.php.

**4.1.3.32** `Binance\API::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 424 of file php-binance-api.php.

**4.1.3.33 BinanceAPI::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 2114 of file php-binance-api.php.

**4.1.3.34 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

**Exceptions**

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

Definition at line 477 of file php-binance-api.php.

**4.1.3.35 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

## Exceptions

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

Definition at line 1003 of file php-binance-api.php.

#### 4.1.3.36 `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 499 of file php-binance-api.php.

#### 4.1.3.37 `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 459 of file php-binance-api.php.

**4.1.3.38 BinanceAPI::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 738 of file php-binance-api.php.

**4.1.3.39 BinanceAPI::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 698 of file php-binance-api.php.

**4.1.3.40 BinanceAPI::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 345 of file php-binance-api.php.

#### 4.1.3.41 **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 362 of file php-binance-api.php.

#### 4.1.3.42 **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 867 of file php-binance-api.php.

#### 4.1.3.43 **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 1643 of file php-binance-api.php.

**4.1.3.44 `BinanceAPI::terminate ( $endpoint )`**

terminate Terminates websocket endpoints.

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

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

**Returns**

null

Definition at line 2002 of file php-binance-api.php.

**4.1.3.45 `BinanceAPI::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 1824 of file php-binance-api.php.

**4.1.3.46 `BinanceAPI::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 559 of file php-binance-api.php.

#### 4.1.3.47 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 1756 of file php-binance-api.php.

**4.1.3.48 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
bool	<i>\$execution_ - callback</i>	callable function

**Returns**

null

**Exceptions**

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

Definition at line 2063 of file php-binance-api.php.

**4.1.3.49 BinanceAPI::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 543 of file php-binance-api.php.



**4.1.3.50** `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**

<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

**Exceptions**

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

Definition at line 597 of file php-binance-api.php.

**4.1.3.51** `Binance\API::withdrawFee ( string $asset )`

withdrawFee get the withdrawal fee for an asset

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

**Parameters**

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

**Returns**

array with error message or array containing withdrawFee

**Exceptions**

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

Definition at line 681 of file php-binance-api.php.

**4.1.3.52** `Binance\API::withdrawHistory ( string $asset = null, array $params = [] )`

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
<i>\$params</i>	array optional startTime, endTime, status parameters

**Returns**

array with error message or array deposit history information

**Exceptions**

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

Definition at line 663 of file php-binance-api.php.

**4.1.4 Member Data Documentation****4.1.4.1 `Binance\API::$api_key` [protected]**

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

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

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

**4.1.4.3 `Binance\API::$base = 'https://api.binance.com/api/'` [protected]**

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

**4.1.4.4 `Binance\API::$chartQueue = []` [protected]**

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

**4.1.4.5 `Binance\API::$charts = []` [protected]**

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

**4.1.4.6 `Binance\API::$curlOpts = []` [protected]**

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

**4.1.4.7 `Binance\API::$depthCache = []` [protected]**

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

**4.1.4.8 `Binance\API::$depthQueue = []` [protected]**

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

**4.1.4.9 `Binance\API::$info` [protected]****Initial value:**

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

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

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

Definition at line 44 of file php-binance-api.php.

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

Definition at line 46 of file php-binance-api.php.

**4.1.4.12** `Binance\API::$stream = 'wss://stream.binance.com:9443/ws/'` `[protected]`

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

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

Definition at line 45 of file php-binance-api.php.

**4.1.4.14** `Binance\API::$wapi = 'https://api.binance.com/wapi/'` `[protected]`

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

# Index

\$api\_key  
    Binance::API, 28  
\$api\_secret  
    Binance::API, 28  
\$base  
    Binance::API, 28  
\$chartQueue  
    Binance::API, 28  
\$charts  
    Binance::API, 28  
\$curlOpts  
    Binance::API, 28  
\$depthCache  
    Binance::API, 28  
\$depthQueue  
    Binance::API, 28  
\$info  
    Binance::API, 28  
\$proxyConf  
    Binance::API, 29  
\$requestCount  
    Binance::API, 29  
\$stream  
    Binance::API, 29  
\$transferred  
    Binance::API, 29  
\$wapi  
    Binance::API, 29  
\_\_construct  
    Binance::API, 10  
\_\_get  
    Binance::API, 10  
\_\_set  
    Binance::API, 10  
  
account  
    Binance::API, 10  
addToTransferred  
    Binance::API, 10  
aggTrades  
    Binance::API, 11  
  
balances  
    Binance::API, 11  
Binance, 5  
Binance::API  
    \$api\_key, 28  
    \$api\_secret, 28  
    \$base, 28  
    \$chartQueue, 28  
    \$charts, 28  
    \$curlOpts, 28  
    \$depthCache, 28  
    \$depthQueue, 28  
    \$info, 28  
    \$proxyConf, 29  
    \$requestCount, 29  
    \$stream, 29  
    \$transferred, 29  
    \$wapi, 29  
    \_\_construct, 10  
    \_\_get, 10  
    \_\_set, 10  
    account, 10  
    addToTransferred, 10  
    aggTrades, 11  
    balances, 11  
    bookPrices, 11  
    buy, 12  
    buyTest, 12  
    cancel, 12  
    candlesticks, 13  
    chart, 13  
    cumulative, 14  
    depositAddress, 14  
    depositHistory, 14  
    depth, 15  
    depthCache, 15  
    displayDepth, 15  
    exchangeInfo, 16  
    first, 16  
    getProxyUriString, 16  
    getRequestCount, 16  
    getTransferred, 17  
    highstock, 17  
    history, 17  
    keepAlive, 17  
    kline, 18  
    last, 18  
    marketBuy, 18  
    marketBuyTest, 19  
    marketSell, 19  
    marketSellTest, 19  
    miniTicker, 20  
    openOrders, 20  
    order, 20  
    orderStatus, 21  
    orders, 21  
    prevDay, 22

- prices, [22](#)
- sell, [22](#)
- sellTest, [23](#)
- setProxy, [23](#)
- sortDepth, [23](#)
- terminate, [24](#)
- ticker, [24](#)
- time, [24](#)
- trades, [24](#)
- useServerTime, [26](#)
- userData, [26](#)
- withdraw, [26](#)
- withdrawFee, [27](#)
- withdrawHistory, [27](#)
- Binance\API, [7](#)
- bookPrices
  - Binance::API, [11](#)
- buy
  - Binance::API, [12](#)
- buyTest
  - Binance::API, [12](#)
- cancel
  - Binance::API, [12](#)
- candlesticks
  - Binance::API, [13](#)
- chart
  - Binance::API, [13](#)
- cumulative
  - Binance::API, [14](#)
- depositAddress
  - Binance::API, [14](#)
- depositHistory
  - Binance::API, [14](#)
- depth
  - Binance::API, [15](#)
- depthCache
  - Binance::API, [15](#)
- displayDepth
  - Binance::API, [15](#)
- exchangeInfo
  - Binance::API, [16](#)
- first
  - Binance::API, [16](#)
- getProxyUriString
  - Binance::API, [16](#)
- getRequestCount
  - Binance::API, [16](#)
- getTransferred
  - Binance::API, [17](#)
- highstock
  - Binance::API, [17](#)
- history
  - Binance::API, [17](#)
- keepAlive
  - Binance::API, [17](#)
- kline
  - Binance::API, [18](#)
- last
  - Binance::API, [18](#)
- marketBuy
  - Binance::API, [18](#)
- marketBuyTest
  - Binance::API, [19](#)
- marketSell
  - Binance::API, [19](#)
- marketSellTest
  - Binance::API, [19](#)
- miniTicker
  - Binance::API, [20](#)
- openOrders
  - Binance::API, [20](#)
- order
  - Binance::API, [20](#)
- orderStatus
  - Binance::API, [21](#)
- orders
  - Binance::API, [21](#)
- prevDay
  - Binance::API, [22](#)
- prices
  - Binance::API, [22](#)
- sell
  - Binance::API, [22](#)
- sellTest
  - Binance::API, [23](#)
- setProxy
  - Binance::API, [23](#)
- sortDepth
  - Binance::API, [23](#)
- terminate
  - Binance::API, [24](#)
- ticker
  - Binance::API, [24](#)
- time
  - Binance::API, [24](#)
- trades
  - Binance::API, [24](#)
- useServerTime
  - Binance::API, [26](#)
- userData
  - Binance::API, [26](#)
- withdraw
  - Binance::API, [26](#)
- withdrawFee
  - Binance::API, [27](#)

withdrawHistory  
    Binance::API, [27](#)