TB API Specifications

General API Information

The base endpoint is:
All endpoints return either a JSON object or array or delimited data with <START><END> between each line (for more efficient data stream with fewer bytes than JSON format (useful for ML).

Data is returned in ascending order. Oldest first, newest last.

All time and timestamp related fields are in milliseconds (using epoch format).

HTTP 10xx return codes are used for for malformed requests; the issue is on the sender's side.

HTTP 20xx return code is used when breaking a request rate limit, and client-sided requests that have errors in it.

HTTP 30xx used for trade related services. (future)

Any API command packets can return an ERROR.

Command end with "j" is for JSON format.
Command end with "d" is for delimited format.

Interval (used for parameter in API commands list):
m -> minutes; h -> hours; d -> days; w -> weeks; M -> months
Interval time formats: 1m, 3m, 5m, 15m, 30m, 1h, 2h, 4h, 6h, 8h, 12h, 1d, 1w, 1M

TB API Specifications

Revisions:

4/20/2021:
Added True Range OCHLV Timeframe interval Examples to help us calculate the OCHLV values based on interval values (1m, 3m, 5m, 15m, 24h, etc.) for all API commands, especially ticker (24h) API command.

4/19/2021:
Added ticker API command with details.

4/3/2021:
Revised the OCHLV intervals description/calculations for timeframe intervals data. 2 charts are added there to be used for reference to help you understand OCHLV data better.

Revised the websocket client details by using wscat command as it supports data streaming/dump for easy troubleshooting. Very important for ML/web pages (using charts/tables).

3/16/2021:
Response json packets for data dump API commands are revised to deal with "k" plus the record index value (Record/Index/Position).

New change: "kn" is used instead of "k". Reason: Easy to parse – and know exact which index (based on record value) in data packet is read to be used for parsing into charts using json.

Example:
"k1", "k2" ….. to end of "kn" records based on index position (k + record). Matches "r" index value in each record/index/position value.

2/8/2021:
Rewritten API Commands to use the standard WebSocket command packet format for JSON.

JSON Specifications: https://jsonapi.org/format/1.1/
https://linuxhint.com/parse-json-data-c/

Example GET request
http (non-SSL)
http://db.networkcities.net:8080/command/parameter1/parameter2/...

https (SSL)
https:// db.networkcities.net:8080/command/parameter1/parameter2/...

See
https://gitlab.com/eidheim/Simple-Web-Server/blob/master/http_examples.cpp
for code reference.

TB API Specifications

API Commands Summary List

Format: command/parameter1/parameter2/…

Individual Kline/Candlestick Data Stream

JSON: klinej/symbol/interval
DELIMITED: klined/symbol/interval

Useful for OCHLV data stream, sending data every interval cycle. Start with Kline Start-end Datetime Range Data dump API command for populating the chart data/ML dataset with OCHLV data dump first, then switch to this API stream command for continuous OCHLV data stream updates. Connection will stay connected as long as the WebSocket connection is active.

Kline/Candlestick Start-end Datetime Range Data Dump

JSON: klinedumpj/symbol/start date/start time/end date/end time
DELIMITED: klinedumpd/symbol/start date/start time/end date/end time

Useful for OCHLV data dump to populate the chart, then switch to data stream for continuous data updates. Very important to utilize the data dump stored to server memory for ML model training.

Ticker (24 hours average) Data Stream

JSON: tickerj/symbol/price
DELIMITED: tickerd/symbol/price

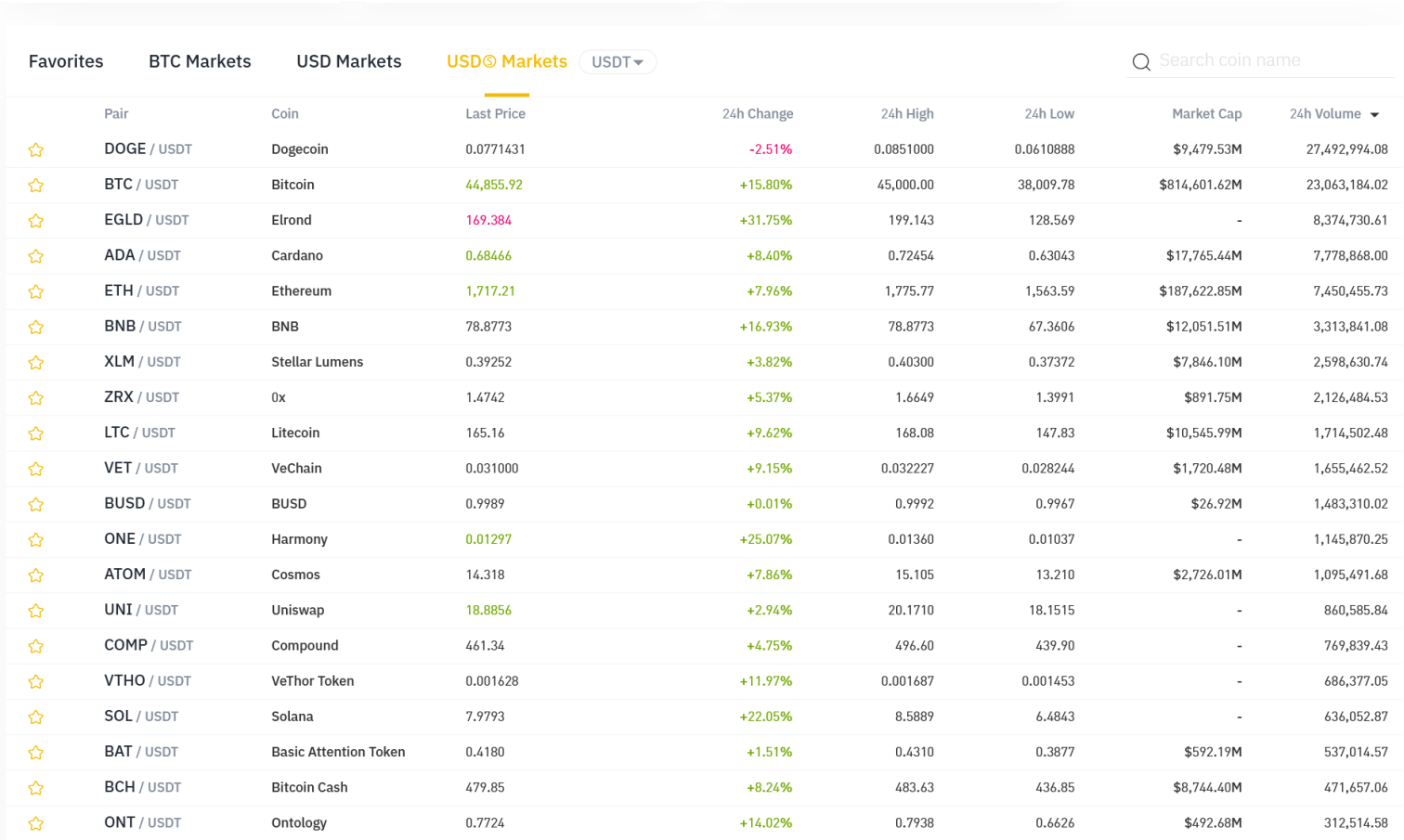
Useful for OCHLV data stream to be used for downloading the top 100 symbols, displaying 24 hours details (change, high, low, volume) and other important data.

2 parts – see below:

1) See this site – look at 24 hours change on top row area:
   https://www.binance.us/en/trade/ADA_USD

TB API Specifications

2) See this site – look at top markets list (based on highest volume as the sort, along with USDT)
https://www.binance.us/en/markets



Favorites | BTC Markets | USD Markets | USDⓈ Markets | USDT | Search coin name

	Pair	Coin	Last Price	24h Change	24h High	24h Low	Market Cap	24h Volume
☆	DOGE / USDT	Dogecoin	0.0771431	-2.51%	0.0851000	0.0610888	$9,479.53M	27,492,994.08
☆	BTC / USDT	Bitcoin	44,855.92	+15.80%	45,000.00	38,009.78	$814,601.62M	23,063,184.02
☆	EGLD / USDT	Elrond	169.384	+31.75%	199.143	128.569	-	8,374,730.61
☆	ADA / USDT	Cardano	0.68466	+8.40%	0.72454	0.63043	$17,765.44M	7,778,868.00
☆	ETH / USDT	Ethereum	1,717.21	+7.96%	1,775.77	1,563.59	$187,622.85M	7,450,455.73
☆	BNB / USDT	BNB	78.8773	+16.93%	78.8773	67.3606	$12,051.51M	3,313,841.08
☆	XLM / USDT	Stellar Lumens	0.39252	+3.82%	0.40300	0.37372	$7,846.10M	2,598,630.74
☆	ZRX / USDT	0x	1.4742	+5.37%	1.6649	1.3991	$891.75M	2,126,484.53
☆	LTC / USDT	Litecoin	165.16	+9.62%	168.08	147.83	$10,545.99M	1,714,502.48
☆	VET / USDT	VeChain	0.031000	+9.15%	0.032227	0.028244	$1,720.48M	1,655,462.52
☆	BUSD / USDT	BUSD	0.9989	+0.01%	0.9992	0.9967	$26.92M	1,483,310.02
☆	ONE / USDT	Harmony	0.01297	+25.07%	0.01360	0.01037	-	1,145,870.25
☆	ATOM / USDT	Cosmos	14.318	+7.86%	15.105	13.210	$2,726.01M	1,095,491.68
☆	UNI / USDT	Uniswap	18.8856	+2.94%	20.1710	18.1515	-	860,585.84
☆	COMP / USDT	Compound	461.34	+4.75%	496.60	439.90	-	769,839.43
☆	VTHO / USDT	VeThor Token	0.001628	+11.97%	0.001687	0.001453	-	686,377.05
☆	SOL / USDT	Solana	7.9793	+22.05%	8.5889	6.4843	-	636,052.87
☆	BAT / USDT	Basic Attention Token	0.4180	+1.51%	0.4310	0.3877	$592.19M	537,014.57
☆	BCH / USDT	Bitcoin Cash	479.85	+8.24%	483.63	436.85	$8,744.40M	471,657.06
☆	ONT / USDT	Ontology	0.7724	+14.02%	0.7938	0.6626	$492.68M	312,514.58

Note: Will add more API commands as I'm working on it. Robert 😊

API Error Codes Response Summary List

Codes	Description
10xx -	Client-Server related errors/issues
20xx -	Invalid data/command
30xx -	Trade errors (for future)

Note: See API Error Codes Response list for more details.

TB API Specifications

NOTE: For Client WebSocket connection to WebSocket server:

1) For the first time Client WebSocket connection to WebSocket server, the WebSocket server will not respond, and will await for client to send the API command packet before sending the response packet, with either response data or error code.

2) If the client WebSocket connection has not sent any API command packets to WebSocket server for the first 60 seconds (timeout timer), the server will automatically terminate the WebSocket connection and will not respond with any response packets.

3) If the client WebSocket connection has already sent command packets to WebSocket server earlier (less than 60 seconds for the first time) and does not disconnect from the WebSocket server for 30 minutes or longer (due to no next API command packets), the WebSocket server will automatically respond with error code packets, then terminate the WebSocket connection to client.

4) To prevent the disconnection by WebSocket server, the client will need to send the API command packet to WebSocket server before the 30 minutes timeout or terminate the WebSocket connection.

The diagrams on the next several pages will help you understand the WebSocket connection and communication methodology.

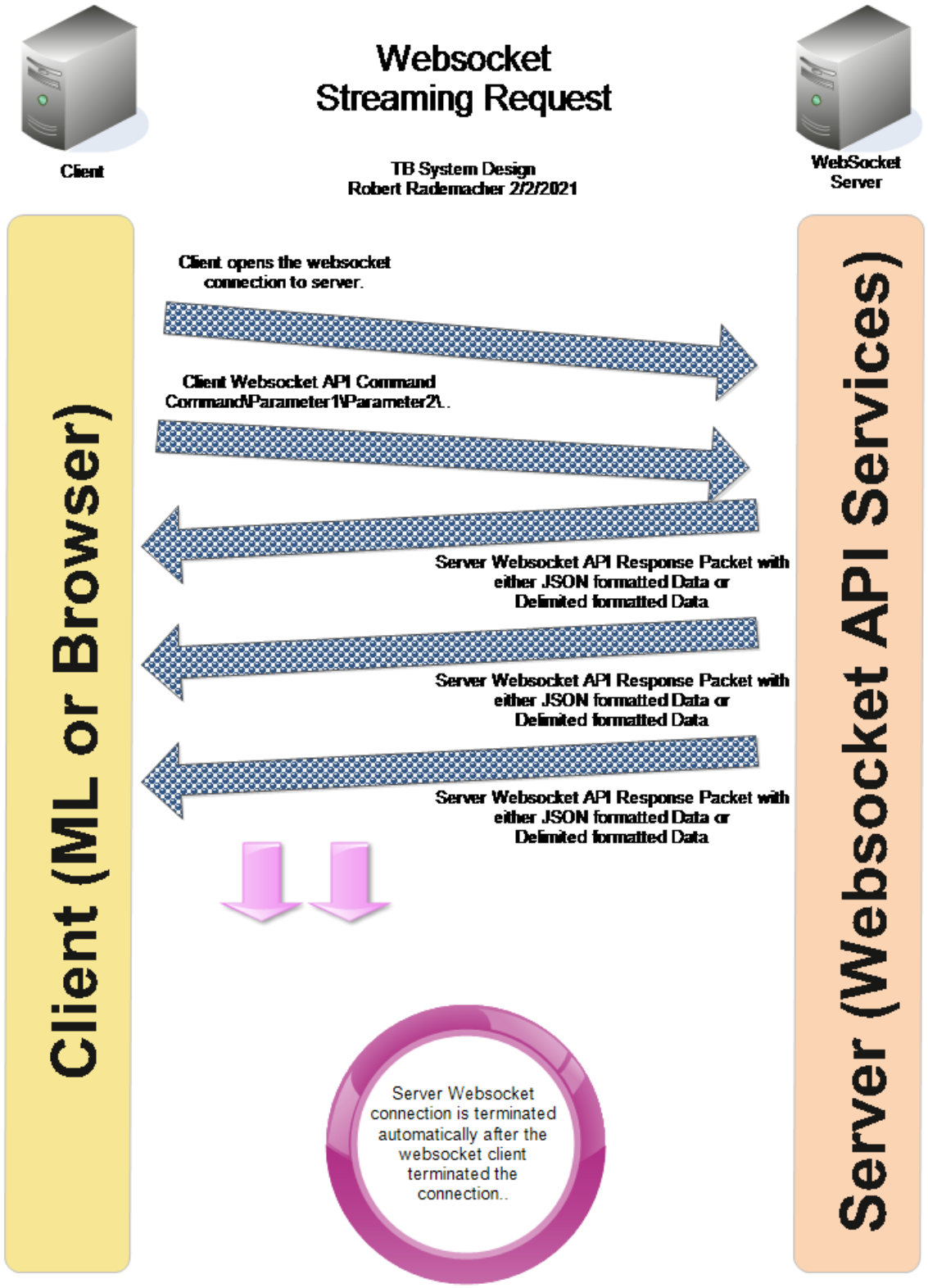
WebSocket Server Code Reference
https://gitlab.com/eidheim/Simple-Web-Server/-/blob/master/server_http.hpp

for reference based on Simple-Web-Server (WebSocket services)
https://gitlab.com/eidheim/Simple-Web-Server

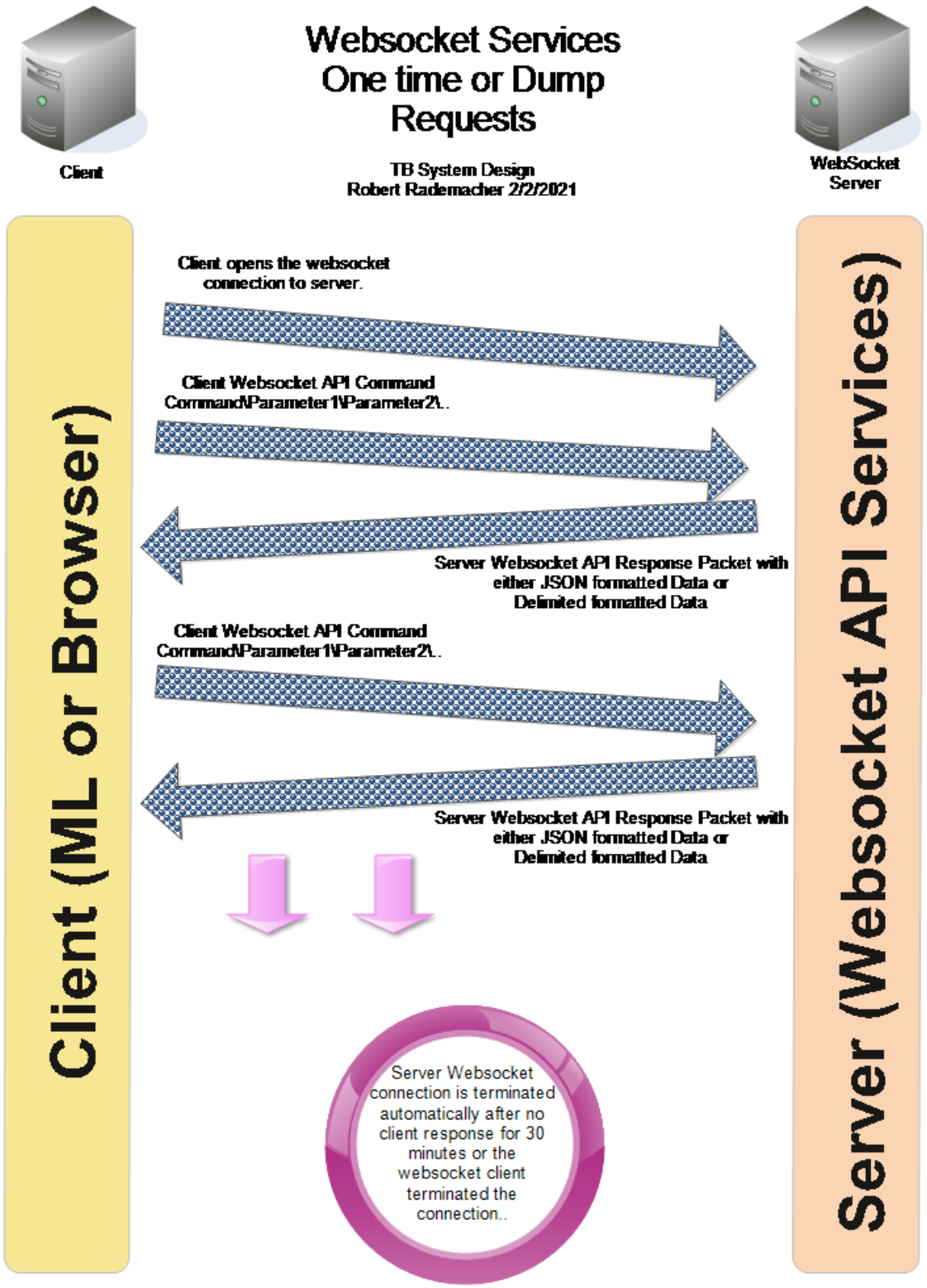
Web-side Client for accessing to WebSocket Server (TESTING)
https://gitlab.com/eidheim/Simple-Web-Server/-/tree/master/web

JSON Websocket Server Examples
https://reposhub.com/cpp/json

TB API Specifications



TB API Specifications



TB API Specifications

True Range OCHLV Timeframe interval details:

Kline/Candlestick chart intervals:
m -> minutes; h -> hours; d -> days; w -> weeks; M -> months
Interval time formats: 1m, 3m, 5m, 15m, 30m, 1h, 2h, 4h, 6h, 8h, 12h, 1d, 1w, 1M

OCHLV Formulas: (using 3m as reference for determining)

Select the first 1m "O" value from these 1m "O" data = Open

Select the last 1m "C" value from these 1m "C" data = Close
n (based on 3m)

Select the highest "H" value from one of these 1m "H" data = High

Select the lowest "L" value from one of these 1m "L" data = Low

1st 1m V + 2nd 1m V + 3rd 1m V + nV…. (n – additional mins > 3) = Total Volume

VERY IMPORTANT:

Keep of count of 1min OCHLV records as they pass so you know when you see a "1m" and a "3m" (or higher) OCHLV bar.

Then:

Set "1m" as the starting High and Low and check of the next two bars to see if the "1m" gets disrupted from either the High or Low position.

For Open is just one of the three OCHLV records, Open on the "1m".

For Close is just one of the three bars, Close on the "3m".

The Open for your three minute OCHLV (3m) is not the Open of 1m but the Close of the just previous 3m bar. This way all the range that happened since the last 3m bar will be included in this new 3m bar.

See 2 charts below on the next 2 pages for your study to help you understand OCHLV bars better.

TB API Specifications

1 Minute chart - look at the first 3 1m bars on left

TB API Specifications

3 Minute chart – see first bar on left, based on 3 1m bars from previous page.

TB API Specifications

True Range OCHLV Timeframe interval Examples:

3m OCHLV

Open = Select the first 1m "O" value from 1st latest 1m "O" records

Close = Select the last 1m "C" value from 3rd latest 1m "C" records

High = Select the highest "H" value from one of 3 (three) latest 1m "H" records

Low = Select the lowest "L" value from one of 3 (three) latest 1m "L" records

Total Volume = 1st 1m V + 2nd 1m V + 3rd 1m V from the entire 3 (three) latest 1m records

60m (OCHLV)

Open = Select the first 1m "O" value from 1st latest 1m "O" records

Close = Select the last 1m "C" value from 60th latest 1m "C" records

High = Select the highest "H" value from one of 60 (sixty) latest 1m "H" records

Low = Select the lowest "L" value from one of 60 (sixty) latest 1m "L" records

Total Volume = 1st 1m V + 2nd 1m V + 3rd 1m V from the entire 60 (sixty) latest 1m records

1d/24h (OCHLV)

Open = Select the first 1m "O" value from 1st latest 1m "O" records

Close = Select the last 1m "C" value from **1440th** latest 1m "C" records

High = Select the highest "H" value from one of **1440** latest 1m "H" records

Low = Select the lowest "L" value from one of **1440** latest 1m "L" records

Total Volume = 1st 1m V + 2nd 1m V + … **1440**th 1m V from the entire **1440** latest 1m records

Note:
1440 minutes is based on **60** minutes x **24** hours calculation.

2d (OCHLV)

Open = Select the first 1m "O" value from 1st latest 1m "O" records

Close = Select the last 1m "C" value from **2880**th latest 1m "C" records

High = Select the highest "H" value from one of **2880** latest 1m "H" records

Low = Select the lowest "L" value from one of **2880** latest 1m "L" records

Total Volume = 1st 1m V + 2nd 1m V + … **2880**th 1m V from **2880** latest 1m records

Note:
2880 minutes is based on **60** minutes x **48** hours calculation.

TB API Specifications

WebSocket Client Details

WebSocket Server port number is 8080.
WebSocket Server Host: https://db.networkcities.net:8080

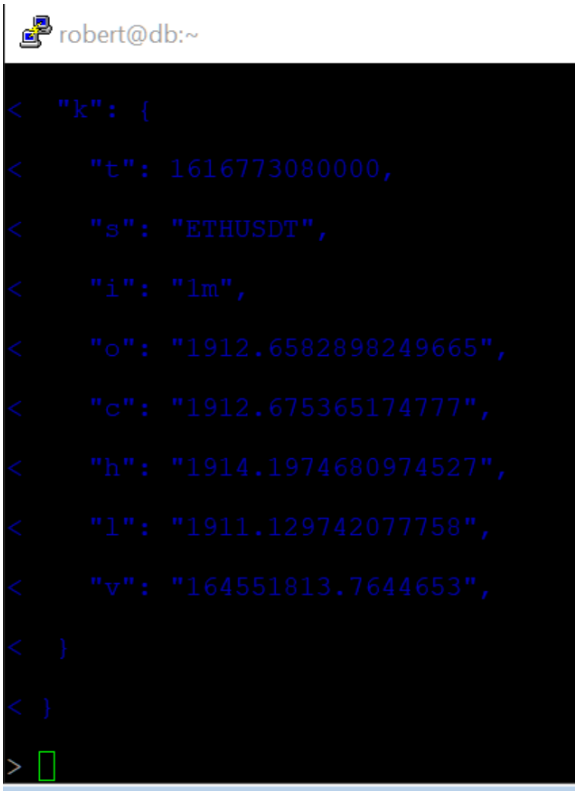
WebSocket Client Communication methods with different programming languages and online command:

Test WebSocket Server using wscat (websocket client tool)

Command line client side test: (for JSON response) for data streaming

wscat -c ws://localhost:8080/klinej/ETHUSDT/3m

Output (inside wscat terminal)

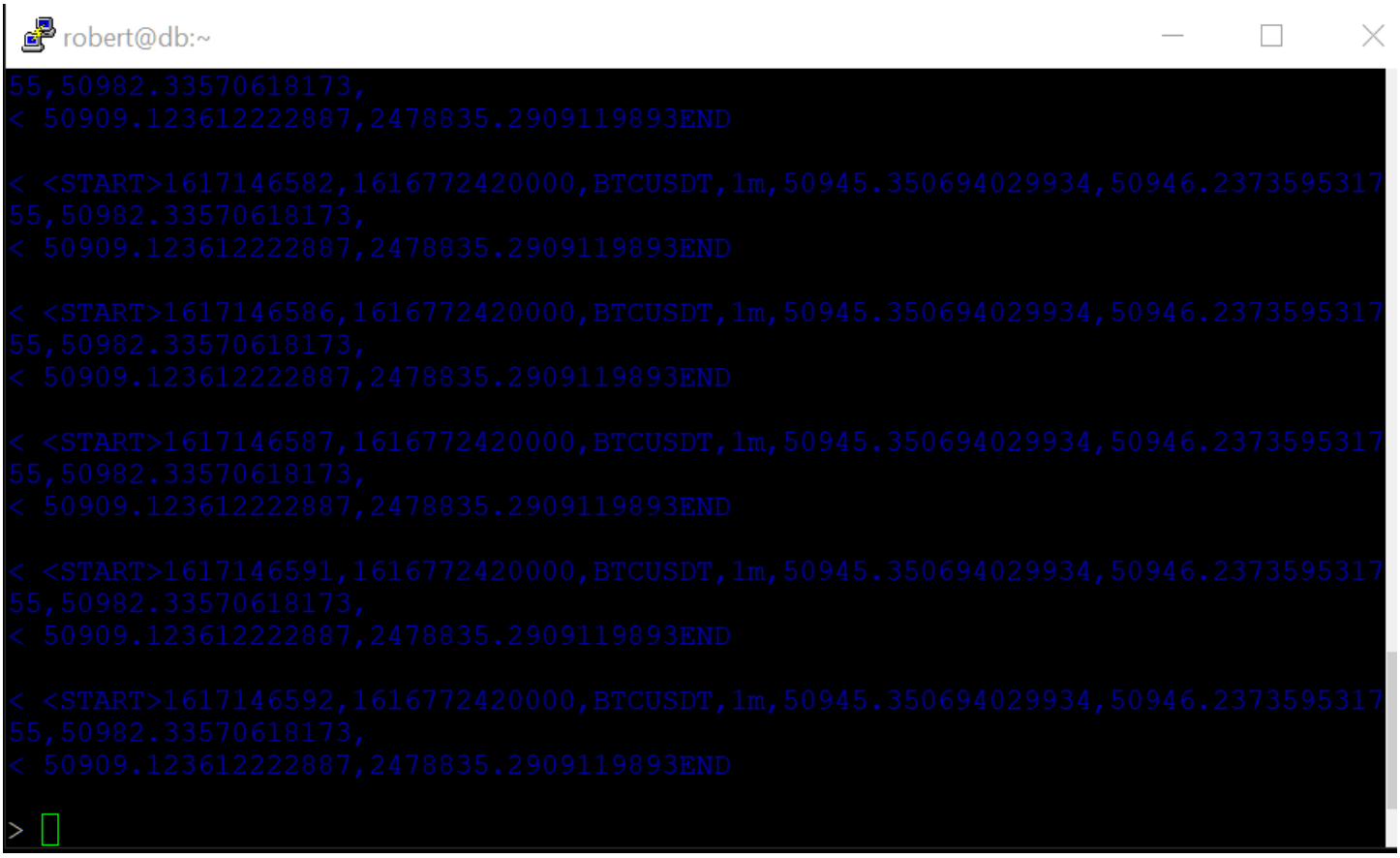


TB API Specifications

Command line client side test (for DELIMITED response) for data dump

wscat -c ws://localhost:8080/klinedumpd/BTCUSDT/1m/2021-01-01/00-00-00/2021-02-01/00-00-00

Output (inside wscat terminal)



Websocket client side examples
https://reqbin.com/req/php/c-dwjszac0/curl-post-json-example (curl test playground)

https://tecadmin.net/post-json-data-with-curl-command/

https://dev.to/ama/curl-commands-examples-to-make-rest-api-calls-4gg3

https://code-boxx.com/send-receive-json-data-php-curl/

https://gist.github.com/ungoldman/11282441

https://medium.com/how-tos-for-coders/https-medium-com-how-tos-for-coders-parse-json-data-using-jq-and-curl-from-command-line-5aa8a05cd79b

TB API Specifications

Test WebSocket Server using PHP (client side)

https://www.w3schools.com/js/js_json_php.asp

PHP JSON Playground
https://loilo.github.io/prettier-php-playground/

Test WebSocket Server using C / C++ (client side)

https://zserge.com/jsmn/

http://lloyd.github.io/yajl/

https://github.com/Tencent/rapidjson

https://linuxhint.com/parse-json-data-c/

https://dzone.com/articles/mapping-json-to-and-from-a-c-structure

WebSocket Client Development Resources:

JSON Tutorial
https://www.tutorialspoint.com/json/index.htm

Websocket side client (JAVASCRIPT JSON) for webpage tutorial
https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON

Websocket side client JSON PHP Reference for webpage:
https://www.tutorialspoint.com/json/json_php_example.htm

Web-side Client for accessing to WebSocket Server
https://gitlab.com/eidheim/Simple-Web-Server/-/tree/master/web

WebSocket Server Development Resources:

WebSocket Server
https://gitlab.com/eidheim/Simple-Web-Server/-/blob/master/server_http.hpp

for reference based on Simple-Web-Server (WebSocket services)
https://gitlab.com/eidheim/Simple-Web-Server

https://dzone.com/articles/mapping-json-to-and-from-a-c-structure

WebSocket Client and Server References:

PHP and HTML5 WebSocket Server and client communication
https://www.cuelogic.com/blog/php-and-html5-websocket-server-and-client-communication

Will add more details here soon. ROBERT 😊

TB API Specifications

API Command Full Details

Kline/Candlestick Data Stream:

The Kline/Candlestick Stream push OCHLV data updates based on interval request every 30 seconds.

Note: The 30 minutes timeout WebSocket server will terminate the data stream connection if there is no update from client side. To keep the WebSocket connection active, client side will need to send the next API command in less than 30 minutes to avoid the disconnection by the WebSocket server.

Frequent API requests (called "spewing") will result in error **code 1003, and will be disconnected by the WebSocket server**.

Kline/Candlestick chart intervals:
m -> minutes; h -> hours; d -> days; w -> weeks; M -> months
Interval time formats: 1m, 3m, 5m, 15m, 30m, 1h, 2h, 4h, 6h, 8h, 12h, 1d, 1w, 1M

Command Packet Formats:
JSON: klinej/symbol/interval
DELIMITED: klined/symbol/interval

Connection status:
Connection will stay connected due to the WebSocket connection – streaming data from WebSocket server to client every 30 seconds with the latest data.

Example:

Command Packet:
http://db.networkcities.net:8080/klinej/BTCUSDT/1m
or
SSL https://db.networkcities.net:8080/klinej/BTCUSDT/1m
Json return data:

```
{
  "e": "kline",     // Event type
  "E": 123456789,   // Event time (today datetime stamp in milliseconds epoch format)
  "s": "BTCUSDT",   // Symbol
  "k": {
    "t": 123400000,    // OCHLV latest datetime epoch format
    "s": "BTCUSDT",    // Symbol
    "i": "1m",         // Interval
    "o": "30000.0010", // Open price
    "c": "30000.0020", // Close price
    "h": "30000.0025", // High price
    "l": "30000.0015", // Low price
    "v": "33240",      // Volume
```

```
  }
}
```

Delimited return data format:
<START>E,t,s,i,o,c,h,l,v<END>

Fields:
E = Event time (today datetime stamp in milliseconds epoch format)
t = OCHLV latest datetime epoch format
s = Symbol
i = Interval
o = Open
c = Close
h = High
l = Low
v = Volume

Example:

Command Packet:
http://db.networkcities.net:8080/klined/ETHUSDT/3m
or
SSL https://db.networkcities.net:8080/klined/ETHUSDT/3m

Delimited return data:
<START>123456789,123300000,ETHUSDT,3m,1010.0010,1010.0020,1011.0025,1013.0015,55640**<END>**

TB API Specifications

Kline/Candlestick Start-end Datetime Range Data Dump:

The Kline/Candlestick Stream push the total records out based on parameters. Useful for populating data on charts, and ML dataset analysis.

Kline/Candlestick chart intervals:
m -> minutes; h -> hours; d -> days; w -> weeks; M -> months
Interval time formats: 1m, 3m, 5m, 15m, 30m, 1h, 2h, 4h, 6h, 8h, 12h, 1d, 1w, 1M

Please read below for the websocket server API services when dealing with the request for the data based on interval time formats:

If the interval time request is 3m (3 minutes), it means that the websocket server API services will pull **3 (THREE) 1m OCHLV records** from database, then use the OCHLV formula to calculate the OCHLV values first before returning the 3m OCHLV values.

Kline/Candlestick chart intervals:
m -> minutes; h -> hours; d -> days; w -> weeks; M -> months
Interval time formats: 1m, 3m, 5m, 15m, 30m, 1h, 2h, 4h, 6h, 8h, 12h, 1d, 1w, 1M

Command Format:
JSON: klinedumpj/symbol/start date/start time/end date/end time
DELIMITED: klinedumpd/symbol/start date/start time/end date/end time

DateTime Range format:
Start Date/Start Time Format: YYYY-MM-DD/HH-MM-SS
End Date/End Time Format: YYYY-MM-DD/HH-MM-SS

Full Command Packet Formats:
http://db.networkcities.net:8080/klinedumpj/SYMBOL/3m/YYYY-MM-DD/HH-MM-SS/YYYY-MM-DD/HH-MM-SS
or
SSL
https://db.networkcities.net:8080/klinedumpj/SYMBOL/3m/YYYY-MM-DD/HH-MM-SS/YYYY-MM-DD/HH-MM-SS

Example:

Command Packet:
http://db.networkcities.net:8080/klinedumpj/BTCISDT/1m/2021-01-01|00-00-00|2021-02-01|00-00-00

Note: Approximately 43,200 records based on 1 month of 1 minute OCHLV records (Total Records Calculation: 60 minutes x 24 hours x 30 days = 43,200). The calculations must be done from WebSocket server side.

Json return data:

```
{
  "e": "klinedumpj",   // Event type
  "E": 123456789,      // Event time (today datetime stamp in milliseconds epoch format)
  "s": "BTCUSDT",      // Symbol
  "a": "43200",        // Total Records
  "k1": {              //k+record (See “r” below)

    "r": 1,            // Record 1 (out of 43200)
    "t": 123400001,    // OCHLV datetime epoch format
    "s": "BTCUSDT",    // Symbol
    "i": "1m",         // Interval
    "o": "30000.0010", // Open price
    "c": "30000.0020", // Close price
    "h": "30000.0025", // High price
    "l": "30000.0015", // Low price
    "v": "33240",      // Volume
  }
  "k2": {              //k+record (See “r” below)
    "r": 2,            // Record 2 (out of 43200)
    "t": 123410002,    // OCHLV datetime epoch format
    "s": "BTCUSDT",    // Symbol
    "i": "1m",         // Interval
    "o": "30000.0010", // Open price
    "c": "30000.0020", // Close price
    "h": "30000.0025", // High price
    "l": "30000.0015", // Low price
    "v": "33240",      // Volume
  }
  "k3": {              // k+record (See “r” below)
    "r": 3,            // Record 3 (out of 43200)
    "t": 123410003,    // OCHLV datetime epoch format
    "s": "BTCUSDT",    // Symbol
    "i": "1m",         // Interval
    "o": "30000.0010", // Open price
    "c": "30000.0020", // Close price
    "h": "30000.0025", // High price
    "l": "30000.0015", // Low price
    "v": "33240",      // Volume
  }
```

```
<clip …displaying 4-4319 records in between here. Bottom – last record>

  "k43200": {           // k+record (See “r” below)
    "r": 43200,        // Record 43200 (out of 43200 - FINAL)
    "t": 123410003,    // OCHLV datetime epoch format
    "s": "BTCUSDT",    // Symbol
    "i": "1m",         // Interval
    "o": "30000.0010", // Open price
    "c": "30000.0020", // Close price
    "h": "30000.0025", // High price
    "l": "30000.0015", // Low price
    "v": "33240",      // Volume
  }
}
```

Note: Approximately 17Mb worth of json return packet based on 1 month of 1 minute OCHLV records (Total Bytes Calculation: 60 minutes x 24 hours x 30 days x 380 bytes = 16,416,000 bytes or 16.Mb).

TB API Specifications

Delimited return data format:
<START>E,a,r,t,s,i,o,c,h,l,v|E,a,r,t,s,i,o,c,h,l,v|…repeat until the end of record<END>

Fields:
E = Event time (today datetime stamp in milliseconds epoch format)
a = Total Records
r = Record/Index/Position
t = OCHLV datetime epoch format
s = Symbol
i = Interval
o = Open
c = Close
h = High
l = Low
v = Volume

Example:

Command Packet:
http://db.networkcities.net:8080/klinedumpd/ETHUSDT/3m/2021-01-01|00-00-00|2021-02-01|00-00-00

Note: Approximately 43,200 records based on 1 month of 1 minute OCHLV records (Total Records Calculation: 60 minutes x 24 hours x 30 days = 43,200).

Delimited return data:
<START>123456789,33240,1,123400001,ETHUSDT,123300000,3m,1010.0010,1010.0020,1011.0025,
1013.0015,55640|123456789,33240,2,123400001,ETHUSDT,123300000,3m,1010.0010,1010.0020,1011.0025,
1013.0015,55640|123456789,33240,3,123400001,ETHUSDT,123300003,3m,1010.0010,1010.0020,1011.0025,
1013.0015,55640|123456789,33240,4,123400001,ETHUSDT,123300006,3m,1010.0010,1010.0020,1011.0025,
1013.0015,55640|
<clip …displaying 4-4319 records in between>
123456789,33240,33240,123400001,ETHUSDT,123333245,3m,1010.0010,1010.0020,1011.0025,
1013.0015,55640<END>

Note: Approximately 4Mb worth of delimited return packet based on 1 month of 1 minute OCHLV records (Total Bytes Calculation: 60 minutes x 24 hours x 30 days x 96 bytes = 4,147,000 bytes or 4Mb).

For ML applications, the delimited format is useful and highly recommended – as it may need 1+ year 1m records (MIN) per crypto pair request (approximately 51Mb data worth of 525,000 1m OCHLV records), compared to json format, which would be around 220Mb.

TB API Specifications

Ticker (24 hours average) Data Stream

JSON: tickerj/symbol/price
DELIMITED: tickerd/symbol/price

Parameters:
1) No Symbol and Price: Send the latest weekly top 100 list with symbols only.
2) Symbol: Display last price, 24h change, 24h percentage change, 24h high, 24h low, 24h volume
3) Symbol AND PRICE: Display last price only.
4) PRICE: Send the latest weekly top 100 list with symbols and price only.
5) Symbol with TOP100: Display last price, 24h change, 24h percentage change, 24h high, 24h low, 24h volume for EVERY SYMBOL (100 RECORDS TOTAL).

The top 100 cryptos are based on 24h volume (ascending) as listed on Binance Markets site: https://www.binance.us/en/markets

Examples (for Parameters 1-5):

#1 Top 100 List
100 weekly symbols dumped from database:
BNBUSDT NEARUSDT LUNAUSDT RSRUSDT SNXUSDT THETAUSDT AUDUSDT XEMUSDT WRXUSDT COMPUSDT ANKRUSDT XTZUSDT XLMUSDT ATOMUSDT AAVEUSDT XRPUPUSDT
DASHUSDT BTCSTUSDT VTHOUSDT LINKUSDT FILUSDT TFUELUSDT JSTUSDT STORJUSDT CHRUSDT RENUSDT TUSDUSDT RUNEUSDT QTUMUSDT USDCUSDT CHZUSDT EOSUSDT FETUSDT
ALGOUSDT COSUSDT ADAUSDT XMRUSDT ONEUSDT BTTUSDT ETCUSDT HBARUSDT ALPHAUSDT RVNUSDT XRPUSDT FLMUSDT AVAXUSDT BANDUSDT BUSDUSDT SOLUSDT IOSTUSDT
DGBUSDT FTTUSDT ETHUSDT SANDUSDT SUSHIUSDT 1INCHUSDT MANAUSDT REEFUSDT CELRUSDT IOTAUSDT CRVUSDT GBPUSDT VETUSDT YFIUSDT SXPUSDT BCHUSDT TRXUSDT
ZECUSDT ZENUSDT NKNUSDT ENJUSDT EGLDUSDT STMXUSDT FTMUSDT DENTUSDT DOTUSDT ZILUSDT OMGUSDT BATUSDT BTCUSDT EURUSDT LTCUSDT NEOUSDT HOTUSDT NPXSUSDT
ICXUSDT UNIUSDT ONTUSDT MATICUSDT SUNUSDT KSMUSDT COTIUSDT WINUSDT DOGEUSDT OGNUSDT WAVESUSDT XVSUSDT MKRUSDT GRTUSDT SCUSDT

JSON Command Packet:
http://db.networkcities.net:8080/tickerj
or
SSL https://db.networkcities.net:8080/tickerj

Json return data: (sorted based on 24h volume)

```
{
  "e": "tickerj",      // Event type
  "E": 1234567891      // Event time (today datetime stamp in milliseconds epoch format)
  "a": "100",          // Total Records
    {                  //
    "r": 1,            // Record 1
    "s": "BNBUSDT",    // Symbol
    },
    {                  //
    "r": 2,            // Record 2
    "s": "NEARUSDT",   // Symbol
    },

Continue sending the json formatted records until the last 100th record:
    {                  //
    "r": 100,          // Record 100
    "s": "SCUSDT",     // Symbol
    },
} //End
```

TB API Specifications

DELIMITED Command Packet:
http://db.networkcities.net:8080/tickerd
or
SSL https://db.networkcities.net:8080/tickerd

DELIMITED return data: (sorted based on 24h volume)

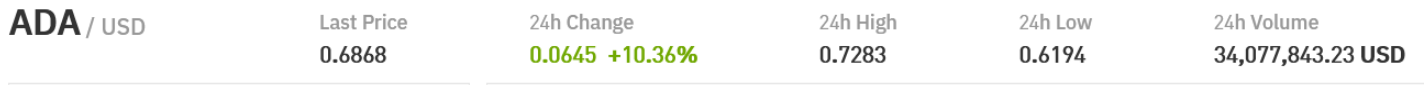
<START>100,BNBUSDT,NEARUSDT,LUNAUSDT,RSRUSDT,SNXUSDT,THETAUSDT,AUDUSDT,XEMUSDT,WRXUSDT,COMPUSDT,ANKRUSDT,XTZUSDT,XLMUSDT,ATOMUSDT,AAVEUSDT,XRPUPUSDT,DASHUSDT,BTCSTUSDT,VTHOUSDT,LINKUSDT,FILUSDT,TFUELUSDT,JSTUSDT,STORJUSDT,CHRUSDT,RENUSDT,TUSDUSDT,RUNEUSDT,QTUMUSDT,USDCUSDT,CHZUSDT,EOSUSDT,FETUSDT,ALGOUSDT,COSUSDT,ADAUSDT,XMRUSDT,ONEUSDT,BTTUSDT,ETCUSDT,HBARUSDT,ALPHAUSDT,RVNUSDT,XRPUSDT,FLMUSDT,AVAXUSDT,BANDUSDT,BUSDUSDT,SOLUSDT,IOSTUSDT,DGBUSDT,FTTUSDT,ETHUSDT,SANDUSDT,SUSHIUSDT,1INCHUSDT,MANAUSDT,REEFUSDT,CELRUSDT,IOTAUSDT,CRVUSDT,GBPUSDT,VETUSDT,YFIUSDT,SXPUSDT,BCHUSDT,TRXUSDT,ZECUSDT,ZENUSDT,NKNUSDT,ENJUSDT,EGLDUSDT,STMXUSDT,FTMUSDT,DENTUSDT,DOTUSDT,ZILUSDT,OMGUSDT,BATUSDT,BTCUSDT,EURUSDT,LTCUSDT,NEOUSDT,HOTUSDT,NPXSUSDT,ICXUSDT,UNIUSDT,ONTUSDT,MATICUSDT,SUNUSDT,KSMUSDT,COTIUSDT,WINUSDT,DOGEUSDT,OGNUSDT,WAVESUSDT,XVSUSDT,MKRUSDT,GRTUSDT,SCUSDT<END>

TB API Specifications

#2 Symbol Parameter

/tickerj/ADAUSDT (see #2 parameter above)

https://www.binance.us/en/trade/ADA_USD



Command Packet:
http://db.networkcities.net:8080/tickerj/ADAUSDT
or
SSL https://db.networkcities.net:8080/tickerj/**ADAUSDT**

Json return data:

```
{
  "e": "tickerj",      // Event type
  "E": 123456789,      // Event time (today datetime stamp in milliseconds epoch format)
  "s": "ADAUSDT",      // Symbol
  "k": {               //k
    "r": 1,            // Record 1
    "t": 123400001,    // OCHLV datetime epoch format
    "s": "ADAUSDT",    // Symbol
    "i": "24h",        // 24 Hour Interval
    "o": "3.0010",     // Last price (recent)
    "c": "3.0020",     // 24h Close price
    "h": "3.0025",     // 24h High price
    "l": "3.0015",     // 24h Low price
    "v": "33240",      // 24h Volume
    “p”: “10.36”      // Change Percentage
  }
```

DELIMITED Command Packet:
http://db.networkcities.net:8080/tickerd/ADAUSDT
or
SSL https://db.networkcities.net:8080/tickerd/ADAUSDT

DELIMITED return data:

<START>ADAUSDT,1,123400001,24h,3.0010,3.0010,3.0010,3.0010,33240,10.36**<END>**

TB API Specifications

#3 Symbol and Price Parameter

/tickerj/BNBUSDT/price (see #3 paramter above)

https://www.binance.us/en/trade/BNB_USD

Command Packet:
http://db.networkcities.net:8080/tickerj/BNBUSDT/**price**
or
SSL https://db.networkcities.net:8080/tickerj/**BNBUSDT**/**price**

Json return data:

```
{
  "e": "tickerj",      // Event type
  "E": 1234567891      // Event time (today datetime stamp in milliseconds epoch format)
  "a": "1",          // Total Records
    {                  //
    "r": 1,            // Record 1
    "s": "BNBUSDT",    // Symbol
    "o": "503.0010",     // Last price (recent)
    },
} //End
```

DELIMITED Command Packet:
http://db.networkcities.net:8080/tickerd/**BNBUSDT**/**price**
or
SSL https://db.networkcities.net:8080/tickerd/**BNBUSDT**/**price**

DELIMITED return data:

<START>BNBUSDT,1,503.0010<END>

TB API Specifications

#4 Top 100 List with Price

100 weekly symbols dumped from database:

BNBUSDT NEARUSDT LUNAUSDT RSRUSDT SNXUSDT THETAUSDT AUDUSDT XEMUSDT WRXUSDT COMPUSDT ANKRUSDT XTZUSDT XLMUSDT ATOMUSDT AAVEUSDT XRPUPUSDT DASHUSDT BTCSTUSDT VTHOUSDT LINKUSDT FILUSDT TFUELUSDT JSTUSDT STORJUSDT CHRUSDT RENUSDT TUSDUSDT RUNEUSDT QTUMUSDT USDCUSDT CHZUSDT EOSUSDT FETUSDT ALGOUSDT COSUSDT ADAUSDT XMRUSDT ONEUSDT BTTUSDT ETCUSDT HBARUSDT ALPHAUSDT RVNUSDT XRPUSDT FLMUSDT AVAXUSDT BANDUSDT BUSDUSDT SOLUSDT IOSTUSDT DGBUSDT FTTUSDT ETHUSDT SANDUSDT SUSHIUSDT 1INCHUSDT MANAUSDT REEFUSDT CELRUSDT IOTAUSDT CRVUSDT GBPUSDT VETUSDT YFIUSDT SXPUSDT BCHUSDT TRXUSDT ZECUSDT ZENUSDT NKNUSDT ENJUSDT EGLDUSDT STMXUSDT FTMUSDT DENTUSDT DOTUSDT ZILUSDT OMGUSDT BATUSDT BTCUSDT EURUSDT LTCUSDT NEOUSDT HOTUSDT NPXSUSDT ICXUSDT UNIUSDT ONTUSDT MATICUSDT SUNUSDT KSMUSDT COTIUSDT WINUSDT DOGEUSDT OGNUSDT WAVESUSDT XVSUSDT MKRUSDT GRTUSDT SCUSDT

Command Packet:

http://db.networkcities.net:8080/tickerj/**price**

or

SSL https://db.networkcities.net:8080/tickerj/**price**

Json return data: (sorted based on 24h volume)

```
{
  "e": "tickerj",      // Event type
  "E": 1234567891      // Event time (today datetime stamp in milliseconds epoch format)
  "a": "100",          // Total Records
    {                  //
    "r": 1,            // Record 1
    "s": "BNBUSDT",    // Symbol
    "o": "503.0010"    // Last price (recent)
    },
    {                  //
    "r": 2,            // Record 2
    "s": "DOGEUSDT",   // Symbol
    "o": "0.4010"      // Last price (recent)
    },
```

Continue sending the json formatted records until the last 100th record:

```
    {                  //
    "r": 100,          // Record 100
    "s": "VETSDT",     // Symbol
    "o": "10.4010"      // Last price (recent)
    },
} //End
```

DELIMITED Command Packet:

http://db.networkcities.net:8080/tickerd/**price**

or

SSL https://db.networkcities.net:8080/tickerd/**price**

DELIMITED return data:

<START>BNBUSDT,1,503.0010|DOGEUSDT,2,503.0010|BTCUSDT,3,50300.10| ..continue..
VETUSDT,100,10.4010|<END>

TB API Specifications

#5 Top 100 List

100 weekly symbols dumped from database:

BNBUSDT NEARUSDT LUNAUSDT RSRUSDT SNXUSDT THETAUSDT AUDUSDT XEMUSDT WRXUSDT COMPUSDT ANKRUSDT XTZUSDT XLMUSDT ATOMUSDT AAVEUSDT XRPUPUSDT DASHUSDT BTCSTUSDT VTHOUSDT LINKUSDT FILUSDT TFUELUSDT JSTUSDT STORJUSDT CHRUSDT RENUSDT TUSDUSDT RUNEUSDT QTUMUSDT USDCUSDT CHZUSDT EOSUSDT FETUSDT ALGOUSDT COSUSDT ADAUSDT XMRUSDT ONEUSDT BTTUSDT ETCUSDT HBARUSDT ALPHAUSDT RVNUSDT XRPUSDT FLMUSDT AVAXUSDT BANDUSDT BUSDUSDT SOLUSDT IOSTUSDT DGBUSDT FTTUSDT ETHUSDT SANDUSDT SUSHIUSDT 1INCHUSDT MANAUSDT REEFUSDT CELRUSDT IOTAUSDT CRVUSDT GBPUSDT VETUSDT YFIUSDT SXPUSDT BCHUSDT TRXUSDT ZECUSDT ZENUSDT NKNUSDT ENJUSDT EGLDUSDT STMXUSDT FTMUSDT DENTUSDT DOTUSDT ZILUSDT OMGUSDT BATUSDT BTCUSDT EURUSDT LTCUSDT NEOUSDT HOTUSDT NPXSUSDT ICXUSDT UNIUSDT ONTUSDT MATICUSDT SUNUSDT KSMUSDT COTIUSDT WINUSDT DOGEUSDT OGNUSDT WAVESUSDT XVSUSDT MKRUSDT GRTUSDT SCUSDT

Command Packet:

http://db.networkcities.net:8080/tickerj/**TOP100**

or

SSL https://db.networkcities.net:8080/tickerj/**TOP100**

Json return data: (sorted based on 24h volume)

```
{
  "e": "tickerj",      // Event type
  "E": 1234567891      // Event time (today datetime stamp in milliseconds epoch format)
  "a": "100",          // Total Records
    {                  //
  "k1": {               //k+Record 1
    "r": 1,            // Record 1
    "t": 123400001,    // OCHLV datetime epoch format
    "s": "ADAUSDT",    // Symbol
    "i": "24h",        // 24 Hour Interval
    "o": "3.0010",     // Last price (recent)
    "c": "3.0020",     // 24h Close price
    "h": "3.0025",     // 24h High price
    "l": "3.0015",     // 24h Low price
    "v": "33240",      // 24h Volume
    “p”: “10.36”      // Change Percentage
    },
  "k2": {               //k+Record 2
    "r": 2,            // Record 2
    "t": 123400001,    // OCHLV datetime epoch format
    "s": "BSBUSDT",    // Symbol
    "i": "24h",        // 24 Hour Interval
    "o": "3.0010",     // Last price (recent)
    "c": "3.0020",     // 24h Close price
    "h": "3.0025",     // 24h High price
    "l": "3.0015",     // 24h Low price
    "v": "33240",      // 24h Volume
    “p”: “10.36”      // Change Percentage
    },
```

Continue sending the json formatted records until the last 100th record:

```
  "k100": {              //k+Record 100
    "r": 1,            // Record 100
    "t": 123400001,    // OCHLV datetime epoch format
    "s": "DOGEUSDT",    // Symbol
    "i": "24h",        // 24 Hour Interval
    "o": "3.0010",     // Last price (recent)
    "c": "3.0020",     // 24h Close price
    "h": "3.0025",     // 24h High price
    "l": "3.0015",     // 24h Low price
    "v": "33240",      // 24h Volume
    “p”: “10.36”      // Change Percentage
    },
} //End
```

DELIMITED Command Packet:

http://db.networkcities.net:8080/tickerd/TOP100

or

SSL https://db.networkcities.net:8080/ticker**d/TOP100**

DELIMITED return data: (sorted based on 24h volume)

<START> ADAUSDT,1,123400001,24h,3.0010,3.0010,3.0010,3.0010,33240,10.36**|**
BSBUSDT,2,123400001,24h,3.0010,3.0010,3.0010,3.0010,33240,10.36**| ..continue..**
DOGEUSDT,100,123400001,24h,0.4010,0.4010,0.4010,0.4010,33240,22.36**|<END>**

API Error Codes Response List

Error code JSON format:

```
{
  "code":xxxx,
  "msg":"Description"
}
```

Error code in DELIMITED format:

<START>code,description**<END>**

Client-Server Services / Network related errors

Codes	Description	Details
1000	– Unknown error	- occurred while processing the client request.
1001	– Disconnected	– Disconnected from server side due to no client response/error.
1002	– Unauthorized	– Client is not authorized to execute the request.
1003	– Too many requests	– Client is spamming the API server with too many requests.
1004	– API server is busy	– Client will need to wait for another request
1007	– Timeout	- Timeout waiting for response from API server.

Data Packet Request related errors

Codes	Description	Details
2000	– Invalid timestamp	- Wrong timestamp format in command packet.
2001	– Illegal characters	– Illegal or wrong format in command packet.
2002	– Unknown request	- Unknown or wrong request in command packet.
2003	– Bad data request	- Data in parameters are bad in command packet.
2004	– Empty parameter	- Missing data in parameter in command packet.
2005	– Bad/unknown symbol	– Unknown or bad symbol in command packet.
2006	– Excessive lookup interval	– Start/End timestamp range is too big
2007	– Bad API key	- API key format is invalid (for security)
2008	– Invalid API key or IP	- Unauthorized access due to wrong API key or IP address

Trade related errors (future)

Codes	Description	Details
30xx	- Trade errors (for future)	