Real-time Crypto Data

Crypto Data API provides websocket streaming for trades, quotes, orderbooks, minute bars and daily bars. This helps receive the most up to date market information that could help your trading strategy to act upon certain market movements.

Since Alpaca now executes your crypto orders in its own exchange, the `v1beta3` crypto market data endpoints no longer distribute data from other providers, but from Alpaca itself.

You can find the general description of the real-time WebSocket Stream here. This page focuses on the crypto stream.

> 👍 **Advanced Websockets Tutorial**
>
> Check out our tutorial Advanced Live Websocket Crypto Data Streams in Python for some tips on handling live crypto data stream in Python.

URL

The URL for the crypto stream is

```
wss://stream.data.alpaca.markets/v1beta3/crypto/us
```

Sandbox URL:

```
wss://stream.data.sandbox.alpaca.markets/v1beta3/crypto/us
```

Multiple data points may arrive in each message received from the server. These data points have the following formats, depending on their type.

Channels

Trades

Schema

Attribute	Type	Notes
T	string	message type, always “t”
S	string	symbol
p	number	trade price
s	number	trade size
t	string	RFC-3339 formatted timestamp with nanosecond precision
i	int	trade ID
tks	string	taker side: B for buyer, S for seller

Example

JSON


```json
{
  "T": "t",
  "S": "AVAX/USD",
  "p": 47.299,
  "s": 29.205707815,
  "t": "2024-03-12T10:27:48.858228144Z",
  "i": 3447222699101865076,
  "tks": "S"
}
```


Quotes

Schema

Attribute	Type	Notes
T	string	message type, always “q”
S	string	symbol
bp	number	bid price
bs	number	bid size
ap	number	ask price

Attribute	Type	Notes
as	number	ask size
t	string	RFC-3339 formatted timestamp with nanosecond precision

Example

JSON


```json
{
  "T": "q",
  "S": "BAT/USD",
  "bp": 0.35718,
  "bs": 13445.46,
  "ap": 0.3581,
  "as": 13561.902,
  "t": "2024-03-12T10:29:43.111588173Z"
}
```


Bars

> 📘 **Crypto bars contain quote mid-prices**
>
> Due to the volatility of some currencies, including lack of trade volume at any given time, we include the quote midpoint prices in the bars to offer a better data experience. If in a bar no trade happens, the volume will be 0, but the prices will be determined by the quote prices.

There are three separate channels where you can stream trade aggregates (bars).

Minute Bars (`bars`)

Minute bars are emitted right after each minute mark. They contain the trades and quote midpoints from the previous minute.

Daily Bars (`dailyBars`)

Daily bars are emitted right after each minute mark after the market opens. The daily bars contain all trades and quote midprices until the time they were emitted.

Updated Bars (`updatedBars`)

Updated bars are emitted after each half-minute mark if a “late” trade arrived after the previous minute mark. For example if a trade with a timestamp of `16:49:59.998` arrived right after

`16:50:00` , just after `16:50:30` an updated bar with `t` set to `16:49:00` will be sent containing that trade, possibly updating the previous bar's closing price and volume.

Schema

Attribute	Type	Description
T	string	message type: "b", "d" or "u"
S	string	symbol
o	number	open price
h	number	high price
l	number	low price
c	number	close price
v	int	volume
t	string	RFC-3339 formatted timestamp

Example

JSON

```json
{
  "T": "b",
  "S": "BTC/USD",
  "o": 71856.1435,
  "h": 71856.1435,
  "l": 71856.1435,
  "c": 71856.1435,
  "v": 0,
  "t": "2024-03-12T10:37:00Z",
  "n": 0,
  "vw": 0
}
```

Orderbooks

Schema

Attribute	Type	Notes
T	string	message type, always "o"

Attribute	Type	Notes
S	string	symbol
t	string	RFC-3339 formatted timestamp with nanosecond precision
b	array	bids: array of p (price) and s pairs. If s is zero, it means that that bid entry was removed from the orderbook. Otherwise it was added or updated.
a	array	asks: array of p (price) and s pairs. If s is zero, it means that that ask entry was removed from the orderbook. Otherwise it was added or updated.
r	boolean	reset: if true, the orderbook message contains the whole server side orderbook. This indicates to the client that they should reset their orderbook. Typically sent as the first message after subscription.

Example

Initial full orderbook

JSON


```json
{
  "T": "o",
  "S": "BTC/USD",
  "t": "2024-03-12T10:38:50.79613221Z",
  "b": [
    {
      "p": 71859.53,
      "s": 0.27994
    },
    {
      "p": 71849.4,
      "s": 0.553986
    },
    {
      "p": 71820.469,
      "s": 0.83495
    },
   ...
  ],
  "a": [
    {
      "p": 71939.7,
      "s": 0.83953
    },
    {
```

```
        "p": 71940.4,
        "s": 0.28025
      },
      {
        "p": 71950.715,
        "s": 0.555928
      },
      ...
    ],
    "r": true
  }
```

`r` is true, meaning that this message contains the whole BTC/USD orderbook. It's truncated here for readability, the actual book has a lot more bids & asks.

Update message

json

```json
{
  "T": "o",
  "S": "MKR/USD",
  "t": "2024-03-12T10:39:39.445492807Z",
  "b": [],
  "a": [
    {
      "p": 2614.587,
      "s": 12.5308
    }
  ]
}
```

This means that the ask price level 2614.587 was changed to 12.5308. If there were previously no 2614.587 ask entry in the orderbook, then it should be added, if there were, its size should be updated.

Remove message

JSON

```json
{
  "T": "o",
  "S": "CRV/USD",
  "t": "2024-03-12T10:39:40.501160019Z",
  "b": [
    {
      "p": 0.7904,
      "s": 0
```

```
    }
  ],
  "a": []
}
```


This means that the 0.7904 bid price level should be removed from the orderbook.

Example

```
$ wscat -c wss://stream.data.alpaca.markets/v1beta3/crypto/us
connected (press CTRL+C to quit)
< [{"T":"success","msg":"connected"}]
> {"action": "auth", "key": "**\***", "secret": "**\***"}
< [{"T":"success","msg":"authenticated"}]
> {"action": "subscribe", "bars": ["BTC/USD"]}
< [{"T":"subscription","trades":[],"quotes":[],"orderbooks":[],"bars":
["BTC/USD"],"updatedBars":[],"dailyBars":[]}]
<
[{"T":"b","S":"BTC/USD","o":26675.04,"h":26695.36,"l":26668.79,"c":26688.7,"v":3.2
03-17T12:28:00Z","n":93,"vw":26679.5912436798}]
<
[{"T":"b","S":"BTC/USD","o":26687.9,"h":26692.91,"l":26628.55,"c":26651.39,"v":11.
03-17T12:29:00Z","n":197,"vw":26651.7679765663}]
```

Updated over 1 year ago

← Real-time Stock Data

Real-time News →

Learn

SDKs & Libraries

GitHub

API Status

Security

About Us

Story

Blog

Contact

We're hiring

Disclosures

Disclosure Library

Form CRS

Form CRS Responses to Conversation Starters

Privacy Policy

Cookie Policy

Terms & Conditions

Submit Feedback

Alpaca

Securities brokerage services are provided by Alpaca Securities LLC ("Alpaca Securities"), member FINRA/SIPC, a wholly-owned subsidiary of AlpacaDB, Inc. Technology and services are offered by AlpacaDB, Inc.

Options trading is not suitable for all investors due to its inherent high risk, which can potentially result in significant losses. Please read Characteristics and Risks of Standardized Options before investing in options.

Cryptocurrency services are made available by Alpaca Crypto LLC ("Alpaca Crypto"), a FinCEN registered money services business (NMLS # 2160858), and a wholly-owned subsidiary of AlpacaDB, Inc. Alpaca Crypto is not a member of SIPC or FINRA. Cryptocurrencies are not stocks and your cryptocurrency investments are not protected by either FDIC or SIPC.

Cryptocurrency is highly speculative in nature, involves a high degree of risks, such as volatile market price swings, market manipulation, flash crashes, and cybersecurity risks. Cryptocurrency regulations are continuously evolving, and it is your responsibility to understand and abide by them. Cryptocurrency trading can lead to large, immediate and permanent loss of financial value. You should have appropriate knowledge and experience before engaging in cryptocurrency trading. For additional information, please click here.

! View Important Disclosures