

# WebSocket Stream



This API provides a [WebSocket](#) stream for real-time market data. This allows you to receive the most up-to-date market information, which can be used to power your trading strategies.

The WebSocket stream provides real-time updates of the following market data:

- [Stocks](#)
- [Crypto](#)
- [Options](#)
- [News](#)

## Steps to use the stream

To use the WebSocket stream follow these steps:

### Connection

To establish a connection use the stream URL depending on the data you'd like to consume. The general schema of the URL is

```
wss://stream.data.alpaca.markets/{version}/{feed}
```

Sandbox URL:

```
wss://stream.data.sandbox.alpaca.markets/{version}/{feed}
```

Any attempt to access a data feed not available for your subscription will result in an error during authentication.



### Test stream

We provide a test stream that is available all the time, even outside market hours, on this URL:

```
wss://stream.data.alpaca.markets/v2/test
```

Use the symbol "FAKEPACA" when trying out this test stream.

Upon successfully connecting, you will receive the welcome message:

JSON

```
[{"T": "success", "msg": "connected"}]
```



### Connection limit

The number of connections to a single endpoint from a user is limited based on the user's subscription, but in most subscriptions (including Algo Trader Plus) this limit is 1. If you try to open a second connection, you'll get this error:

JSON

```
[{"T": "error", "code": 406, "msg": "connection limit exceeded"}]
```

## Authentication

You need to authenticate yourself using your credentials. This can be done multiple ways

### For the Trading API, Authenticate with HTTP headers

You can set the same headers used for the historical market data and trading endpoints:

- APCA-API-KEY-ID
- APCA-API-SECRET-KEY

Here's an example using a WebSocket client called [websocat](#):

```
$ websocat wss://stream.data.alpaca.markets/v2/test \
-H="APCA-API-KEY-ID: {KEY_ID}" -H="APCA-API-SECRET-KEY: {SECRET}"
```

### For the Broker API, Authenticate with Basic Authentication

You can use the same Basic Authentication header used for the historical market data and trading endpoints:

- Authorization = base64encode({KEY}:{SECRET})

**Note:** base64encode({KEY\_ID}:{SECRET}) is the base64 encoding of the {KEY}:{SECRET} string.

### For both Trading & Broker API, Authenticate with a message

Alternatively, for both the trading & broker API, you can authenticate with a message after connection:

JSON

```
{"action": "auth", "key": "{KEY_ID}", "secret": "{SECRET}"}
```

Keep in mind though, that you only have 10 seconds to do so after connecting.

If you provided correct credentials you will receive another success message:

JSON

```
[{"T": "success", "msg": "authenticated"}]
```

## Subscription

Congratulations, you are ready to receive real-time market data!

You can send one or more subscription messages. The general format of the subscribe message is this:

JSON

```
{
  "action": "subscribe",
  "<channel1>": ["<SYMBOL1>"],
  "<channel2>": ["<SYMBOL2>", "<SYMBOL3>"],
  "<channel3>": ["*"]
}
```

You can subscribe to a particular symbol or to every symbol using the `*` wildcard. A subscribe message should contain what subscription you want to add to your current subscriptions in your session so you don't have to send what you're already subscribed to.

For example in the test stream, you can send this message:

JSON

```
{"action": "subscribe", "trades": ["FAKEPACA"]}
```

The available channels are described for each streaming endpoints separately.

Much like subscribe you can also send an unsubscribe message that subtracts the list of subscriptions specified from your current set of subscriptions.

JSON

```
{"action": "unsubscribe", "quotes": ["FAKEPACA"]}
```

After subscribing or unsubscribing you will receive a message that describes your current list of subscriptions.

JSON

```
[{"T": "subscription", "trades": ["AAPL"], "quotes": ["AMD", "CLDR"], "bars": ["*"], "updatedBars": [], "dail
```

You will always receive your entire list of subscriptions, as illustrated by the sample communication excerpt below:

JSON

```
> {"action": "subscribe", "trades": ["AAPL"], "quotes": ["AMD", "CLDR"], "bars": ["*"]}
< [{"T": "subscription", "trades": ["AAPL"], "quotes": ["AMD", "CLDR"], "bars": ["*"], "updatedBars": [], "dail
...
> {"action": "unsubscribe", "bars": ["*"]}
< [{"T": "subscription", "trades": ["AAPL"], "quotes": ["AMD", "CLDR"], "bars": [], "updatedBars": [], "dail
```

# Messages

## Format

Every message you receive from the server will be in the format:

```
json
[{"T": "{message_type}", {contents}}, ...]
```

Control messages (i.e. where `T` is `error`, `success` or `subscription`) always arrive in arrays of size one to make their processing easier.

Data points however may arrive in arrays that have a length that is greater than one. This is to facilitate clients whose connection is not fast enough to handle data points sent one by one. Our server buffers the outgoing messages but slow clients may get disconnected if their buffer becomes full.

## Content type

You can use the `Content-Type` header to switch between text and binary message [data frame](#):

- `Content-Type: application/json`
- `Content-Type: application/msgpack`


## Encoding and Compression

Messages over the websocket are encoded as clear text.

To reduce bandwidth requirements we have implemented compression as per [RFC-7692](#). [Our SDKs](#) handle this for you so in most cases you won't have to implement anything yourself.

## Errors

You may receive an error during your session. Below are the general errors you may run into.

Code	Message	Description
400	invalid syntax	The message you sent to the server did not follow the specification.  This can also be sent if the symbol in your subscription message is in invalid format.
401	not authenticated	You have attempted to subscribe or unsubscribe before <a href="#">authentication</a> .
402	auth failed	You have provided invalid authentication credentials.
403	already authenticated	You have already successfully authenticated during your current session.
404	auth timeout	You failed to successfully authenticate after connecting. You only have a few seconds to authenticate after connecting.
405	symbol limit exceeded	The symbol subscription request you sent would put you over the limit set by your subscription package. If this happens your symbol subscriptions are the same as

Code	Message	Description
		they were before you sent the request that failed.
406	connection limit exceeded	You already have the number of sessions allowed by your subscription.
407	slow client	You may receive this if you are too slow to process the messages sent by the server. Please note that this is not guaranteed to arrive before you are disconnected to avoid keeping slow connections active forever.
409	insufficient subscription	You have attempted to access a data source not available in your subscription package.
410	invalid subscribe action for this feed	You tried to subscribe to channels not available in the stream, for example to <code>bars</code> in the <a href="#">option stream</a> or to <code>trades</code> in the <a href="#">news stream</a> .
500	internal error	An unexpected error occurred on our end. Please let us know if this happens.

Beside these there can be some endpoint specific errors, for example in the [option stream](#).

## Example

Here's a complete example of the test stream using the [wscat](#) cli tool:

JSON

```
$ wscat -c wss://stream.data.alpaca.markets/v2/test
Connected (press CTRL+C to quit)
< [{"T":"success","msg":"connected"}]
> {"action":"auth","key":"<YOUR API KEY>","secret":"<YOUR API SECRET>"}
< [{"T":"success","msg":"authenticated"}]
> {"action":"subscribe","bars":["FAKEPACA"],"quotes":["FAKEPACA"]}
< [{"T":"subscription","trades":[],"quotes":["FAKEPACA"],"bars":["FAKEPACA"]}
< [{"T":"q","S":"FAKEPACA","bx":"0","bp":133.85,"bs":4,"ax":"R","ap":135.77,"as":5,"c":["R"],"z":
< [{"T":"q","S":"FAKEPACA","bx":"0","bp":133.85,"bs":4,"ax":"R","ap":135.77,"as":5,"c":["R"],"z":
< [{"T":"b","S":"FAKEPACA","o":132.65,"h":136,"l":132.12,"c":134.65,"v":205,"t":"2024-07-24T07:56
```

 Updated 8 months ago

Did this page help you? ☐ Yes ☐ No