FAQ

Can't find the answer you're looking for? We've shared some of our most frequently asked questions to help you out!

Table of Contents

Cloud

How can I connect to the streaming API?

If you are using a dataset offered through our streaming API, note that the streaming API is accessible over the public internet, so there's no additional network settings needed; if you are running in AWS the data will travel over AWS' private network.

When using the streaming API, do I need to have an account on AWS?

No, you can run the streaming API in your own data center. You would go to the following GitHub links which provide instructions on how to download either the Java SDK or Python SDK.

Java: https://github.com/Nasdaq/NasdaqCloudDataService-SDK-Java
Python: https://github.com/Nasdaq/NasdaqCloudDataService-SDK-Python

What is the benefit of having an AWS account?

The benefit of using AWS is that it will have the lowest latency since it's through the AWS private backbone (Amazon to Amazon).

Latency Statistics: https://www.cloudping.co/grid

Are there additional fees associated with using AWS?

AWS bills customers directly to take the data out of the system, however, these fees are significantly cheaper since it's AWS to AWS.

What is the difference in latency between cloud service and direct connectivity?

Collocated is in the order of 10-20 microseconds while the streaming API is 100-300 milliseconds.

What's the typical latency for data feeds?

We define our latency SLA from when the data is created off the matching engine to when the data is available for consumption at our perimeter (not when it appears in the consumer's system, as we can't control the networking, connectivity, etc. elements once the data leaves our perimeter).

Our SLA is around 300 milliseconds for when the data is available for consumption.

If our system is running in AWS Hong Kong, what's the typical latency?

The AWS latency just from US East 1 > AP East 1 could be 200-250ms plus the streaming API latency on top of that (see previous question).

https://www.cloudping.co/grid/p_50/timeframe/1M

Can I run the streaming API service over Wi-Fi?

You can use Wi-Fi to use the streaming API for basic development (i.e. to check out a feed, etc.), however, if you want to run a big product (i.e. Smart Options, etc.), it might flood your network.

Kafka

What is Kafka?

Kafka is an open source software which provides a framework for storing, reading and analyzing streaming data. It is designed to be run in a "distributed" environment, which means that rather than sitting on one user's computer, it runs across several servers, leveraging the additional processing power and storage capacity that this brings.

How long do you store data in Kafka on a topic?

We store data on Kafka topic for 24 hours.

When do you suggest we start-up our consumer?

We recommend that you start up your consumer at the beginning of the day and close the consumer at the end of the day according to the table below. Starting and closing your consumers daily will mitigate the chances of transient errors due to general maintenance.

System	Feeds	Startup Time (EST)	Shutdown time (EST)
Nasdaq Equities	Totalview Itch	3:00am	8:05pm
	QBBO	3:00am	8:05pm
BX Equities	Totalview Itch	3:05am	6:45pm
	QBBO	3:05am	6:45pm
	Last Sale	3:05am	7:05pm
PSX Equities	Totalview Itch	3:05am	5:05pm
	QBBO	3:05am	5:05pm
	Last Sale	3:05am	5:05pm
NLS+	NLS+ CTA	2:00am	8:30pm
	NLS+ UTP	2:00am	8:30pm
UF	CTA/A	12:00am	8:35pm
	CTA/B	12:00am	8:35pm
	UTP	12:00am	8:35pm
	GIDS	Sunday 7:10pm Monday 7:10pm ... Thursday 7:10pm	Monday 6:30pm Tuesday 6:30pm ... Friday 6:30pm
MTNewswires		24/7	
Benzinga		24/7	

Nasdaq Fund Network		24/7	
ORPA UF	Smart Options	OPRA has 2 sessions. A pre-market "Extended Hours" session, published on channels 91->94, and then a "Regular Hours" session, published on channels 1->48. OPRA publishes on the following schedules: Extended Hours: 02:30->09:15 Regular Hours: 06:10->19:00 We run our system with wider start and end times. Extended Hours: running 00:03->10:35, subscribing to upstream feeds from 02:30:00->10:00:00 Regular Hours: running 00:03->19:35, subscribing to upstream feeds from 06:05:00->19:05:00	

How can I get the earliest or latest data?

For SDK versions 0.4.x and higher:

Setting the auto.offset.reset property in kafka-config.properties to latest, will start the consumer from where it last stopped consuming or failed (not with latest message).

Setting the auto.offset.reset property in kafka-config.properties to earliest, will always start the consumer reading data after midnight (12:00:00 AM ET)

The default value for auto.offset.reset is earliest.

For SDK versions 0.3.x and lower:

This is done by setting the consumer's `auto.offset.reset` property when connecting. See the complete documentation: https://kafka.apache.org/0100/javadoc/constant-values.html#org.apache.kafka.clients.consumer.ConsumerConfig.AUTO_OFFSET_RESET_CONFIG

You can either connect to the topic from `earliest` (default) or `latest` when you pass in the `auto.offset.reset` property in https://github.com/Nasdaq/CloudDataService/blob/master/ncdssdk-client/src/main/resources/kafka-config.properties

    auto.offset.reset=earliest

- or -

    auto.offset.reset=latest

What is suggested cluster size for the Mirrormaker environment to ensure low latency?

We recommend deploying a Mirrormaker instance for each topic getting mirrored to your kafka cluster. 1 CPU and 8 GB of memory for each instance of Mirrormaker should be sufficient.

What is the suggested cluster size for a Kafka Cluster managed by AWS MKS?

There are many factors/parameters to be considered for sizing the Kafka clusters which include the number of topics, the number of consumers/producers connecting to topics, the number of replicas, how the leader partition for each topic are distributed within the cluster, among other considerations.

Nasdaq Data Link currently has 25 Kafka brokers, where each Kafka broker is of R5.xlarge size and 500 GB of storage. We have different incoming/outgoing feed requirements on Nasdaq Data Link than you may have in consuming GIDS and Smart Options.

For our Kafka Cluster (mirrored) what are partition and replication-factor specifications for each topic? We would like to maintain specs similar to Nasdaq Data Link cluster specs.

The streaming API uses 3 replicas for each topic and a 24 hour retention period on each topic.

Are there "optimal" kafka configuration settings that you, or your customers, have found to improved performance?

It depends on how fast you are able to process the data, max.poll.records, max.partition.fetch.bytes and max.poll.interval.ms are all related. We experimented with two settings to improve performance:

- max.poll.records=5000 (default=500)
    - As you adjust max.poll.records to a larger value, your kafka consumer should be able to commit offset within the max.poll.interval.ms value. If your kafka consumer does not "keep up," then you will get kafka consumer warning message.
- fetch.max.bytes=52428800*4 (default=1)
- max.partition.fetch.bytes=5242880 (default=1048576)

We have also included recommended base configurations in the Python and Java SDK (link to the config files). Please tune the following configurations to your system, if needed, based on this article **here**.

Are the topic schemas likely to change overtime? How would you suggest we maintain a schema registry?

Real-time data feeds rarely change schema doesn't change, possibly once a year at most. Schemas are written daily to Kafka topics named 'control'.

If you have similar suggestions on sizes of cluster for s3 and PostgreSQL connector cluster, please advise.

The streaming API doesn't use an S3 connector, but rather uses custom consumers (one per topic) that write data to S3; each consumer uses 0.5 CPU and 2 GB of memory.

Would you recommend converting message value using the schema before storing in S3 or PostgreSQL sinks?

It really depends on your storage pattern and partitioning scheme. Messages on Kafka topics are serialized and stored in Avro format, directly from Kafka without deserializing. If you need to partition data by some type (messagetype, symbol, etc), then you'll need to deserialize the message. Nasdaq Data Link converts the Avro messages to Parquet in a consumer before writing to S3.

How can I get messages as quickly as possible? I don't need to acknowledge or anything.

You can use async commit for polling data quickly but there is drawback; you might want to take a look at this thread before applying https://stackoverflow.com/questions/46546174/kafka-consumer-commitsync- vs-commitasync this property. Here's an example of using async commit: https://github.com/Nasdaq/CloudDataService/blob/master/ncdssdk- client/src/main/java/com/nasdaq/ncdsclient/NCDSSession.java#L300

What are helpful metrics to troubleshoot Kafka connection issues?

Plotting the following metrics (with fetch-latency-avg and fetch-latency-max as the time period) would be helpful in identifying issues relating to unexpected disconnections by your consumer:

- fetch-latency-avg: The average time taken for a fetch request.

- fetch-latency-max: The max time taken for a fetch request.

- fetch-rate: The number of fetch requests per second.

- fetch-size-avg: The average number of bytes fetched per request.

- fetch-size-max: The maximum number of bytes fetched per request.

- fetch-throttle-time-avg: The average throttle time in ms. When quotas are enabled, the broker may delay fetch requests in order to throttle a consumer which has exceeded its limit. This metric indicates how throttling time has been added to fetch requests on average.

- fetch-throttle-time-max: The maximum throttle time in ms.

Here is a resource that further details Kafka consumer metrics.

SDK

What languages do you offer SDKs for?

Currently, we officially maintain the Python and Java SDKs:

- Java SDK documentation: https://github.com/Nasdaq/NasdaqCloudDataService-SDK-Java
- Python SDK documentation: https://github.com/Nasdaq/NasdaqCloudDataService-SDK-Python

However, other languages are in the works. The streaming API platform is based on Kafka and avro message format, so there are a variety of other community supported SDKs:

- C++ https://github.com/edenhill/librdkafka
- .NET https://github.com/confluentinc/confluent-kafka-dotnet

The streaming API uses Oauth for authentication, so you would have to add that flow to all three of these examples. For the .NET example, you would need to add a method for avro derealization.

Do you offer filters on the data for a message type?

Yes, see the example https://github.com/Nasdaq/CloudDataService#example-syntax-to-run-the-client-jar- based-on-this-sdk

11. To get all messages for given type message type

java -jar ncdssdk-client.jar -opt GETALLMSGS -topic NLSUTP -msgName SeqTradeReportMessage

I'm running the SDK locally and timing out? Any suggestions?

Depending on the feed volumes, you could be saturating your network. In the SDK, you can try changing the timeout setting variable in the following classes:

https://github.com/Nasdaq/CloudDataService/blob/master/ncds- sdk/src/main/java/com/nasdaq/ncdsclient/internal/ReadSchemaTopic.java#L38

https://github.com/Nasdaq/CloudDataService/blob/master/ncds- sdk/src/main/java/com/nasdaq/ncdsclient/internal/ReadSchemaTopic.java#L96

try increasing this variable to 20 seconds:

```
Duration sec = Duration.ofSeconds(10);
```

Do you offer filters on the data, for a symbol universe?

No, all the data feeds are firehose-style, so you get all the messages. For example, if you are consuming GIDS feed - you'll want to apply a filter on the Intraday Index Value (IIV) Messages and possibly the Equities Summary Messages to parse out the .IXIC symbol. If you use our SDK, then the pseudo code would look like:

```
for (ConsumerRecord<String, GenericRecord> record : records) { Schema.Field symbolField =
record.value().getSchema().getField("instrumentID"); String sym = null;
    if (symbolField != null) {
        sym = ((org.apache.avro.util.Utf8) record.value().get(symbolField.pos())).toString().trim();
    }
        if ((symbols == null || symbolSet.contains(sym)) && (msgTypes == null
|| msgTypeSet.contains(msg_t))) {
            // put logic here System.out.println(record.value().toString());
        }
}
```

and you would plug the filtering code into
https://github.com/Nasdaq/CloudDataService/blob/master/ncdssdk-
client/src/main/java/com/nasdaq/ncdsclient/NCDSSession.java#L264

Certificates

What's the recommended approach to certificate management (either in SDK or customer code)?

We use trusted SSL certificate authority for communication with the Nasdaq Data Link system. You don't need to pin certificates on your system.

If you are using the Java or Python SDK with pinning certificate, we request you to move to new version of SDKs. For java starting from 0.5.0 and python 0.2.0.

Security

We need to whitelist the streaming API ingress/egress on our firewall. What are the IP's and ports?

The streaming API runs in AWS US-EAST-1. Refer to AWS's guide https://docs.aws.amazon.com/general/latest/gr/aws-ip-ranges.html.
The streaming API runs on ports 443 (auth/auz) and 9094 (kakfa).

Partnership

We would like to partner with Nasdaq Data Link to distribute our real-time data? What information do you need to kick off the engagement?

To start the effort, we'll need details around the following areas:

- Data & Messages
    - average number of messages per day
    - average size of message
    - 1 millisecond peak message rate (in last 6 months)
    - 1 second per peak rate (in last 6 months)
- Data Pipeline
    - Do you operate in the cloud or in own datacenter?
    - Do you already send data to the cloud?
    - What protocols and message formats are in use (ITCH, SOUP, FIX, etc)?
    - Is the expectation that they will push the data to Nasdaq Data Link or will Nasdaq Data Link fetch data from their system?

Data

What are NextShares ("M") messages?

For NextShares please see the following [link](#).

Does Nasdaq Data Link maintain a schema registry (ie. a confluent schema registry)?

Nasdaq Data Link does not maintain a schema registry, but does maintain schemas in the control topic. At the start of each market day, we push the schema for the data feeds from the `StreamInitiated` message to the control topic. You can maintain your schema registry by reading data from the control topic. Schemas rarely update – generally once every few years.

For Nasdaq Basic, what's the mapping of topics to symbols?

Nasdaq Basic (QBBO) is delivered via three topics:

Topic	Securities Coverage
QBBO-A-CORE	NYSE-listed issues
QBBO-B-CORE	NYSE MKT-, NYSE Arca-, BATS-, and U.S. regional stock exchange- listed issues
QBBO-C-CORE	Nasdaq-listed issues

For Nasdaq Last Sale Plus, what's the mapping of topics to symbols?

NLS+ is delivered via two topics:

Topic	Trades Coverage
NLSUTP	Nasdaq trades with real-time consolidated volume for Nasdaq-listed securities
NLSCTA	Nasdaq trades with real-time consolidated volume for NYSE-, NYSE MKT-, NYSE Arca, BATS-, and other listed securities

For Smart Options, what's the mapping of topics to symbols?

There are 48 channels for market hours and 4 channels for extended hours. See pages 80 and 81 of the OPRA spec for what symbols are on what channels https://assets.website-files.com/5ba40927ac854d8c97bc92d7/6022f04ccc39cd2fc1220c48_OPRA%20Binary%20DR%20Spec%20Version%204.3_020521.pdf

Topic names attached:

smart-options-topics.
txt

How do you recommend consuming the Smart Options feed (ie. via Kafka consumer, Mirror- Maker, etc.)?

It depends on your use case. If multiple downstream systems depend on data, then using mirror-maker would be better approach, as you can have multiple systems consume the data at once from you your kafka system.

For Historical CQT, to automate the file download, what time should we set the download for?

You can call the API periodically (ie. every 15 minutes) checking for the request_status value and download the nighty file when request_status = Complete. The times vary daily due to market volumes but usually available by 4AM EST.

What's the best way to work with message sequence numbers to ensure order and completeness?

Our recommendation for handling/processing with message sequence is as follows:

On beginning of day (one system event messages): Reset variable to 0;
    (processMessageNumber=0)
On every other message:
    If(messageSequnceNumber <= processMessageNumber) { Ignore message;
    }
    Else If (messageSequnceNumber > processMessageNumber+1) {
        LOG WARN (GAP DATA);
    }
    Else {
        processMessageNumber = messageSequnceNumber
        ProcessMessage(BusinessLogic)
    }

We recommend this approach as Kafka (or any distributed system) can have dupes. We are working on reducing this from happening, but it can happen given the distributed nature of Kafka.

How do I request a specific sequence id?

The SDK doesn't expose a method for requesting a specific sequence id, but a user can request by to start the stream from a timestamp. Also, kafka itself keeps track of the last message read by a consumer ("consumer offset"), so a consumer can restart with the latest offset or a specific
offset. https://kafka.apache.org/0100/javadoc/constant-values.html#org.apache.kafka.clients.consumer.ConsumerConfig.AUTO_OFFSET_RESET_CONFIG

What is the difference in latency between cloud service and being collocated?

Collocated is in the order of 10-20 microseconds while the streaming API is 100-300 milliseconds.

What's the typical latency for datafeeds?

We define our latency SLA from when the data is created off the matching engine to when the data is available for consumption at our perimeter, not when it appears in the consumer's system, as we can't control the networking, connectivity, etc elements once the data leaves our perimeter. Our SLA is around 300 milliseconds for when the data is available for consumption.

How do I get the timestamp?

There's a timestamp on the message from kafka and a timestamp inside the message from the exchange itself.

In kafka, each ConsumerRecord object has a timestamp property on the message when the message is written to the kafka topic.

In each message, the trackingID column is a compound key containing:

- bytes 0-1 = Nasdaq internal tracking number
- bytes 2-7 = Timestamp in nanoseconds from midnight

See the TrackingID class in the com.nasdaq.ncdsclient.internal.utils package within the SDK for parsing the timestamp inside each message.

How do I get consolidated volume using Nasdaq Basic and NLS+?

Nasdaq Basic is a real-time data feed with trades being reports from the Nasdaq, BX, PSX, and two Nasdaq TRF's.

There are two options to get the consolidated data from the Nasdaq Basic feed:

1. Trade Report message from
   :http://www.nasdaqtrader.com/content/technicalsupport/specifications/dataproducts/NLSPlus-   cloud.pdf
   1. Consolidated Volume: Reflects the volume for the Issue Symbol as reported on the consolidated market data feed at the time that the trade message was generated. See Appendix A for calculation and dissemination information for this data element.
2. EOD Trade Summary Message:
   http://www.nasdaqtrader.com/content/technicalsupport/specifications/dataproducts/NLSPlus-   cloud.pdf
   1. Consolidated Volume: Consolidated Volume long Reflects the total volume for the issue reported at the consolidated market level.

The Trade Report Message is appended to each real-time Trade update, so if the want is to have consolidated data in a more real-time manner, that may be the best way.

For accessing historical data, what is the process to start copying bulk paraquet data from your S3 to my AWS account?

We store historical data in AWS S3 buckets, so you'll need to provide us your AWS IAM Role for us to grant access to AWS S3 buckets.

If our system is running in AWS Hong Kong, what's the typical latency?

The AWS latency just from US East 1 > AP East 1 could be 200-250ms plus the streaming API latency on top of that (see previous question).

https://www.cloudping.co/grid/p_50/timeframe/1M

What is the transmission schedule for the data feeds?

You can find the transmission schedule here.

For Smart Options, is it normal to receive NBBO messages (i.e. For TWO_SIDED_NBBO_SHORT) with zero bid and ask price?

This is expected and possible behavior for the feed and indicates that there is currently no market or NBBO for this option, this update indicates, there was a market, but now it's been removed and NBBO is 0 x 0 at the moment.

http://www.nasdaqtrader.com/Trader.aspx?id=DPSpecs