User Guide to Cloud Stream API

Jana Credé

22.05.2024
Leipzig

Ref. 4.1

Table of Contents

1 Introduction

The Cloud Stream API allows customers to easily retrieve real-time public market data feeds of selected EEX products via cloud-based solution. The solution is designed to minimize the entry barrier for market data access by using standard solutions and technologies:

- Connectivity is available via public internet by using WebSocket technology
- Both binary (Google Protocol Buffers – GPB) and ascii (JSON) data encoding is supported
- Message content is – as far as possible – aligned to FIX specification; thus, field names and valid values can be easily understood and in addition, supported valid values are documented in the GPB protocol description.
- Cloud Stream API is furthermore designed for support any kind of data feeds, beside trades, quotes, aggregates (L2) orderbook also incremental data feeds with full depth orderbook data.

This document describes the message layouts and the technical details of the streaming solution.
https://github.com/Deutsche-Boerse/Cloud.Stream.Client/tree/main/proto/src/client.proto

This manual relates to the interface version number 001.000.006.

2 Authentication and Authorization

The streaming WebSocket API is secured by using standard Bearer authorization method and a short living access token – which is only required during establishing the connection. The token can be generated by the following URL:

```
https://md.deutsche-boerse.com/login
```

The POST request has to provide the username / password as JSON object in the body part:

```
{
      “username”: “customer_id”,
      “password”: “secure_pwd”
}
```

In addition, the HTTP Header field “Content-Type” has to be set to json:

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

The response will contain beside the access token some additional fields; for accessing the stream, only the access token field is relevant:

```
{
      ”AccessToken”: “<token>”
      “ExpiresIn”: <seconds>
      “TokenType”: “Bearer”
}
```

The real-time stream can be reached by the following URL:

```
wss://md.deutsche-boerse.com/stream?format=<json|proto>
```

The access token has to be provided in the HTTP Authorization header field:

```
Authorization: Bearer <token>
```

Differences between json and binary format:

You can choose during connecting to the feed between binary and json format. Depending on the targeted performance, binary format is faster to process and consumes less bandwidth, than json, and is to be preferred in general.

3 Messaging – feed subscription

https://github.com/Deutsche-Boerse/Cloud.Stream.Client/tree/main/proto/src/client.proto

In order to retrieve market data feed, the customer has to select the interested data stream. For this reason, after the physical connection has been established, customer has to *subscribe* to the stream of interest. This can be done sending the Subscribe request to the server:

Fieldname	Description	Content
event	Event (action) name	“subscribe”
requestId	Optional request id, will be returned to the client as part of the response message.	Identification provided by the customer.

stream	List of stream name and stream offset in case of data recovery.
	`“md-microproducts”`
	`“md-tradegate”`
	(see Appendices)
		e.g.
		`{“stream”: “md-microproducts”, “startTime”: “1659008334123456789”}`
		startTime is in nanoseconds since epoch, TZ=UTC
		or
		`{“stream”: “md-microproducts”, “startSeq”: 12345}`
		startSeq is an integer to be used to start at the first message having the sequence number or the next one available

Unsubscribe

If the customer is no more interested to receive data from the stream, the unsubscribe request can be used:

Fieldname	Description	Content
event	Event (action) name	“unsubscribe”
requestId	Optional request id, will be returned to the client as part of the response message.	Identification provided by the customer.

stream	List of stream names.
	`“md-microproducts”`
	`“md-tradegate”`
	(see Appendices)
		e.g.
		`“stream”: [“md-microproducts”]`

Example

```
{"event":"subscribe", "requestId":123456789, "subscribe":{"stream":[{"stream": "md-tradegate"}]}}
{"event":"unsubscribe", "requestId":123456789, "unsubscribe":{"stream":["md-tradegate"]}}
```

The Request to Subscribe or Unsubscribe will be replied with Response containing a Status field that will inform about the result.

The count and order of Response messages is the same as the one in repeated streams field of the Request.

Client is disconnected in case the Request is malformed.

Example for Status=OK

```
{"subs": "md-tradegate", "messages": [{"@type": "type.googleapis.com/Client.Response", "requestId":
"123456789"}]}
```

4 Service availability

The service will be technically available 24x7 without interruption; planned maintenance will be announced in advance. The respective data made available via Cloud Stream is only available and updated if the relevant trading venue or other source(s) of the data is open for business and/or trading.

Service	Telephone	Email
Functional Support	+49 (0)69 – 211 - 11540	datafeeds@deutsche-boerse.com
Contractual Support	+49-(0)341 – 2156 - 288	datasource@eex-group.com

5 Data and service messages

The messages which will be sent to the customer based on his subscription were defined in a generic mode so that any kind of messages can be transported by the same technical interface. The StreamMessage contains the following fields:

Field Name	Type	Description
subs	string	Subscription information – stream or topic name
seq	uint64	Sequence number of the message in the stream; this number has to be used on case of message recovery.
messages	object	List of messages – the list will contain usually only a single message.

5.1 Google Protocol Buffers

The usage of Google Protocol Buffers is forcing some standard. Please be recommended to read https://protobuf.dev/programming-guides/proto3/#default for the usage of default values. E.g. values for enum fields are not sent, if they are default values (‘0’=).

5.2 JSON Format

The same is valid for using JSON format. Please be recommended to read https://protobuf.dev/programming-guides/proto3/#json for the usage of JSON format.

6 Appendix A

n/a

7 Appendix B

n/a

8 Appendix C

n/a

9 Appendix D

n/a

10 Appendix E – EEX Gas Spot description

https://github.com/Deutsche-Boerse/Cloud.Stream.Client/tree/main/proto/src/md_energy.proto

version number 001.000.007

For Gas Spot the stream specification in the subscribe and unsubscribe has to be detailed in the following way:

Stream / Topic name	Description

energy_gas_<market-area>	Stream name for orders and trades. Market-area can be one of the following:
	cegh-vtp, cz-vtp, etf, nbp, peg, pvb, the, ttf, ztp, others.
	
	cegh-vtp = Austria
	cz-vtp = Czech Republic
	etf = Denmark
	nbp = United Kingdom
	peg = France
	pvb = Spain
	the = Germany
	ttf = Netherlands
	ztp = Belgium
	others = else, any other

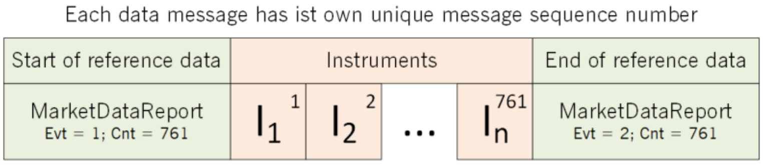
energy_gas_stl_<market-area>	Stream name for settlements. Market-area can be one of the following:
	cegh-vtp, cz-vtp, etf, nbp, peg, pvb, the, ttf, ztp, others.
	
	cegh-vtp = Austria
	cz-vtp = Czech Republic
	etf = Denmark
	nbp = United Kingdom
	peg = France
	pvb = Spain
	the = Germany
	ttf = Netherlands
	ztp = Belgium
	others = else, any other

energy_gas_reference	(common) Reference data stream for all market-areas.

10.1 General reference data processing rules

A snapshot cycle consists of (see Figure 1):

- A market data report message (Evt = 1 = “StartOfReferenceData”).
- A sequence of a data messages of type instrument
- Finally, market data report message (Evt = 2 = “EndOfReferenceData”).



Each data message has ist own unique message sequence number

Start of reference data	Instruments				End of reference data
MarketDataReport Evt = 1; Cnt = 761	I_1 [1]	I_2 [2]	...	I_n [761]	MarketDataReport Evt = 2; Cnt = 761

Figure 1 Entire snapshot cycle of the Energy Gas Spot reference data stream

with:

I_x: Instrument x

This reference data snapshot cycle will be repeated every 60 minutes and repeated cycles may contain more (in case of newly added) or less (in case of expired) instruments. For this reason, customer's application should start receiving the reference data stream e.g. 60 minutes in the past, wait for the 'StartOfReferenceData' message which will also contain the total number of messages as part of the reference data cycle; then receive all reference data information and finish the process by receiving the 'EndOfReferenceData' information.

10.2 Message structure description

The tables below contain detailed description of each message type of the Energy Gas Spot feed:

- Reference data messages: MarketDataReport, Instrument
- Market data messages: Order, Trade
- Market data messages: Settlement

10.2.1 Message type: Market Data Report

FIX Tag	FIX Field Name	Req'd	FAST Data Type	Description
	Evt	Y	enum	Event 0 = UNKNOWN_EVENT 1 = START_OF_REFERENCE_DATA 2 = END_OF_REFERENCE_DATA
	Cnt	Y	int32	total message count of the current reference data cycle

10.2.2 Message type: Instrument

FIX Tag	FIX Field Name	Req'd	FAST Data Type	Description

	> Seq	Y	int32	Seq (technical sequence)
48	> ID	Y	int64	SecurityID
41251	> Base	Y	string	StreamCommodityBase
41252	> CmdtyType	N	string	StreamCommodityType
41255	> Desc	Y	string	StreamCommodityDescription
41055	> Start	Y	uint64	DeliveryScheduleSettlStart
41056	> End	N	uint64	DeliveryScheduleSettlEnd
41266	> Period	N	string	StreamCommoditiyNearbySettlDayPeriod
41258	> UOM	N	string	StreamCommodityUnitOfMeasure
41253	> SIID	N	int32	StreamCommoditySecurityID-InstID
41253	> SSID	N	int32	StreamCommoditySecurityID-SequenceID
41253	> FIID	N	int32	StreamCommoditySecurityID-FirstSequenceItemID
41253	> SQID	N	int32	StreamCommoditySecurityID-SecondSequenceItemID
41257	> EncDesc	N	string	EncodedStreamCommodityDescription
341	> StartTm	N	uint64	Trading Session Start Time
345	> EndTm	N	uint64	Trading Session End Time

10.2.3 Message type: Order

FIX Tag	FIX Field Name	Req'd	FAST Data Type	Description
48	ID	Y	int64	Identifier of the Instrument
278	MDID	Y	string	MDEntryID
279	UpdtAct	Y	enum	MDUpdateAction '0' = NEW '1' = CHANGE '2' = DELETE '6' = QUERY

270	Px	N	double	Price – MDEntryPrice
271	Sz	N	double	Size – MDEntrySize
269	Typ	Y	enum	MDEntryType ‘0’ = BID ‘1’ = OFFER ‘2’ = BOTH
1070	MDQteTyp	N	enum	MDQuoteType ‘0’ = INDICATIVE ‘5’ = FIRM ‘6’ = WITHHELD ‘7’ = REFERENCE ‘8’ = ALL
59	TmInForce	N	enum	TimeInForce ‘0’ = DAY ‘1’ = GOOD_TILL_CANCEL ‘4’ = FILL_OR_KILL ‘6’ = GOOD_TILL_DATE ‘13’ = FILL_AND_KILL ‘14’ = MARKET_ORDER
144	ImpldMktInd	N	enum	ImpliedMarketIndicator ‘0’ = NOT_IMPLIED ‘5’ = NATIVE ’6’ = EXTERNAL ‘7’ = ALL ‘8’ = VENUE
18	ExecInst	N	enum	ExecutionInstruction ‘0’ = STAY_ON_OFFER_SIDE ‘G’ = ALL_OR_NONE
273	Tm	N	uint64	MDEntryTime

10.2.4 Message type: Trade

FIX Tag	FIX Field Name	Req'd	FAST Data Type	Description
48	ID	Y	int64	Identifier of the Instrument
278	MDID	Y	string	MDEntryID
279	UpdtAct	Y	enum	MDUpdateAction '0' = NEW '1' = CHANGE '2' = DELETE '6' = QUERY
270	Px	N	double	Price – MDEntryPrice
271	Sz	N	double	Size – MDEntrySize
273	Tm	N	uint64	MDEntryTime

10.2.5 Message type: Settlement

FIX Tag	FIX Field Name	Req'd	FAST Data Type	Description
48	ID	Y	int64	Identifier of the Instrument
730	SetlPx	N	double	SettlPrice
273	Tm	N	uint64	SettlTime

10.3 Good to know

10.3.1 Order books in gas spot markets

Orders will automatically be removed from the order books:

- at 02:00 CET for all Within-Day orders
- at 03:00 CET on the last trading day for all Day-Ahead, Weekend, Saturday, Sunday and Individual day and Hourly product orders.

For the NBP market area (United Kingdom), those times are shifted by +1 hour.

At these times, which correspond to the end of the trading period for the concerned contracts or, in other words, when the contracts expire, **orders will be definitively deleted, instead of set to "withheld"** as it might be known from the past. If order shall be maintained, they must be re-entered in the order book. By this principle, compliance is ensured that each order refers to an individual contract. Therefore, spot orders cannot remain in the order book beyond the time period associated with the underlying contract, nor can these orders be transferred from one contract to another.

Hence you can assume, that starting at these times the order books for the corresponding contracts are empty and are started to be filled again for the next day's contract set.

The same should be valid, if there is any maintenance window when the market operator must delete all prices as soon as the products are closed.

If the order is left as "withheld" the price owner can update it after the re-opening if the contract hasn't expired yet.

Of course, the corresponding orders with delete action will be sent at these times 02:00:00 CET, 03:00:00 CET (resp. 03:00:00 CET and 04:00:00 CET for NBP) or immediately before.

11 Change log

No	Chapter, page	Date	Change
1.0 – 3.1			n/a
4.0	Appendix E	September 29, 2023	Added Appendix E for EEX Gas Spot
4.0.1	Appendix E	February 12, 2024	Added 10.3 for order books in EEX Gas Spot
4.1	Appendix E	May 22, 2024	Addition of settlement price to CloudStream