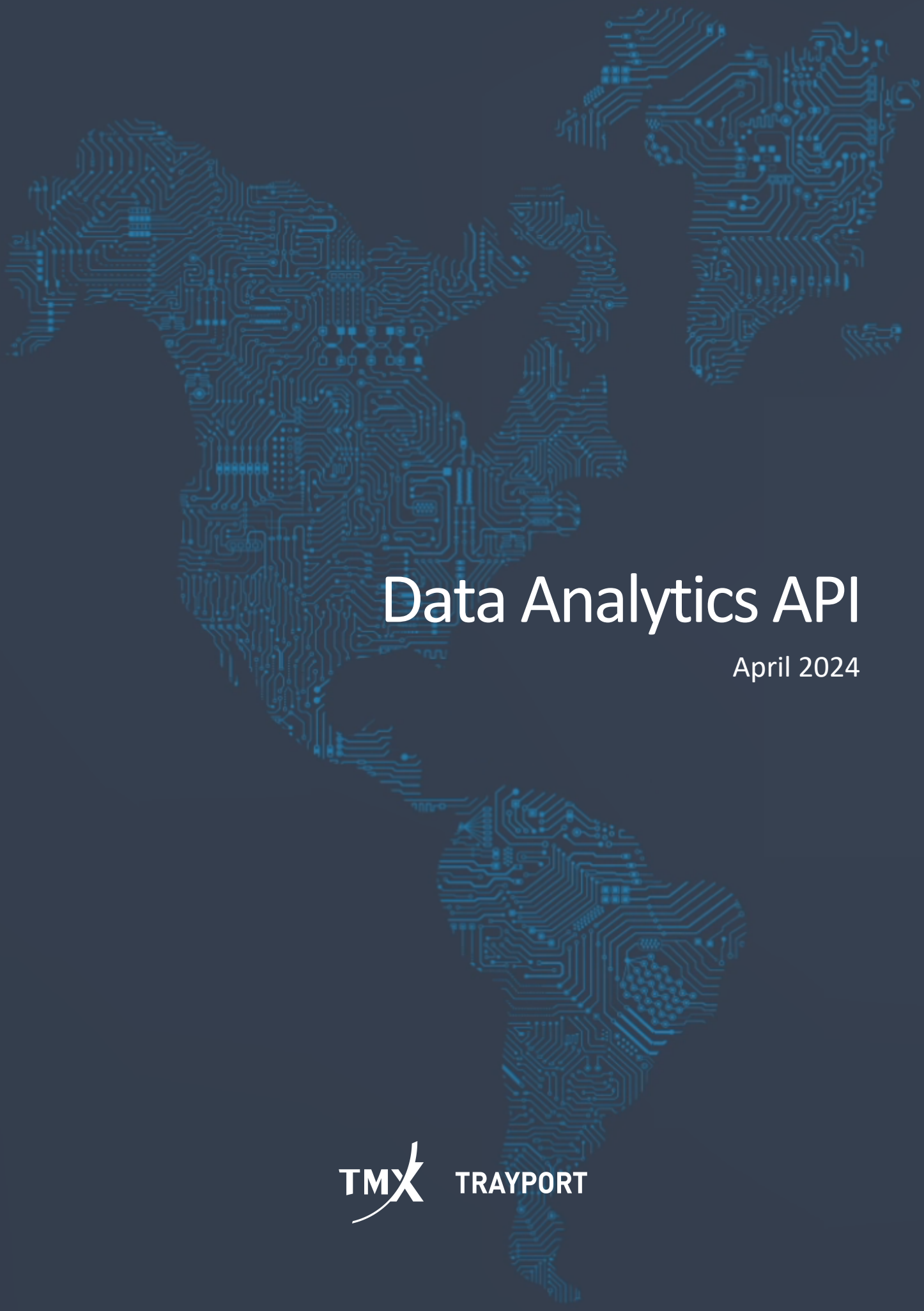


Data Analytics API

April 2024

TMX TRAYPORT

LEGAL NOTICE

7th Floor, 9 Appold Street, London, EC2A 2AP

Trayport Home Page

CONTENTS

DATA ANALYTICS API

The Data Analytics API allows you to retrieve historical order and trade events beyond what is available in Joule or the Joule Direct API. All trades that have ever been made visible to you through Trayport's systems are available, as well as all orderbooks back to 2020.

For more information about the data available to you, see Available Data.

USING THE API

The Analytics API is a REST API. Requests are made and responses are received over HTTPS. To use it, issue a GET request to one of the API endpoints described below using URL parameters to provide information about the data you want to retrieve.

Authentication is provided by sending an API key as a header to the request. For more information about authentication, rate limits and other technical details, see Technical Overview.

The base URL for the Analytics API is:

https://analytics.trayport.com/api

Swagger documentation for the Analytics API can be found here:

https://www.trayport.com/en/support/daapi/index.html#/

To access the Swagger documentation, you need a Trayport account. If you do not have a Trayport account or do not know your username, please contact your Trayport Client Relationship Manager to request this.

AVAILABLE ENDPOINTS

The following endpoints are available in the Data Analytics API and are described in more detail in the sections below.

ORDERS

- **Level 1** (/orders/book/top) - Returns the price and quantity of the best bid and ask for each time interval.
- **Level 2** (/orders/book) - Returns details of orders for each time interval up to the specified maximum depth (10 by default). Private orders (i.e. those belonging to companies over which the user has "identify" permissions) can be optionally identified.

TRADES

- **Trades** (/trades) - Returns details of individual trades for the time interval. Private trades (i.e. those belonging to companies over which the user has "identify" permissions) can be optionally identified.
- **OHLCV** (/trades/ohlcv) - Returns the Open, High, Low, Close and Volume (and optionally VWAP) for each time interval.
- **Last** (/trades/last) - Returns the time and price of the last trade for the given contract. An optional parameter can be used to retrieve the last price at any given point in time.
- **Activity** (/trades/activity) - Returns a list of unique contracts traded in the given time period and the number of trades for each.
- **Private Trades** (/trades/private) - Returns a list of trades belonging exclusively to companies over which the user has "identify" permissions for the time interval.

SNAPSHOTS

- **Snapshots** (/snapshots) - Returns the price and quantity of the best bid and ask at a particular point in time for a given set of contracts. It also returns the time and price of the last trade for the respective contracts.

Apart from /trades/activity, all these endpoints share a common set of parameters for specifying the contract for which data is retrieved. These endpoints support querying a single contract; the parameters are described in Specifying a Contract. The /trades, /trades/private, and /snapshots endpoints support various ways to specify multiple contracts. These are described under each respective endpoint.

Level 1 (/orders/book/top)

Returns a timeseries of the best bid and ask orders in the market for the given contract.

The following parameters are required:

- Contract specifiers as described in Specifying a Contract.
- **from** and **until**: The dates and times between which order data should be returned. Provide these as ISO-8601 datetimes with no fractional seconds; for example, "2023-03-22T12:34.56Z". Lower-precision timestamps are accepted and rounded to the start of the next largest time unit; for example, "2023-03-22" is interpreted as "2023-03-22T00:00:00Z". Similarly, UTC is assumed if you do not specify a time zone.

  The time range you can request is limited to the shorter of 60 days or 100,000 points of the specified interval. If you attempt to query a range larger than this you will receive a "400" (bad request) response code.
- **intervalUnit** and **interval**: Used to define the interval between each data point in the time series. For example, if you wanted to retrieve the state of the best bid and ask every 15 seconds between your given **from** and **until** times, you would set the **intervalUnit** to "second" and the **interval** to "15". The following values are valid for **intervalUnit**:
  - second
  - minute
  - hour.

  interval must be a whole number greater than or equal to one.

  The smallest time interval that can be requested is 1 second; it is not currently possible to view the orderbook at a higher resolution than this. If you would like Trayport to develop this feature, please contact your Trayport Client Relationship Manager to register your interest.

The following parameters may also be used but are not required:

- **includeEmptyBuckets**: Whether time intervals in which there were no bids or asks in the market should be included in the response. If this parameter is not specified it will assume a default value of “false” and times at which there were no orders in the market will not be included in the response. Valid values are “true” or “false”.
- **routes** as described in Routes to Market.
- **optionalFields**: A list of additional fields to be included in the response. The following additional fields may be requested:
  - **route** and **routeId** as described in Routes to Market.
  - **venueCode**: The Trayport-defined code for the broker or exchange on which the order was placed. A list of these codes can be found in Venue Codes. These

values will be inserted into the response as two fields called **bidVenueCode** and **askVenueCode**.

For example, to retrieve the best bids and asks every fifteen seconds from 09:00 to 16:00 (UTC) on 1st March 2023 for NBP Apr-23, you could run the following query:

```
https://analytics.trayport.com/api/orders/book/top?from=2023-03-01T09&until=2023-03-01T16&interval=15&intervalUnit=second&instrumentId=10002071&sequenceId=10000305&sequenceItemId=232&contractType=SinglePeriod
```

The response will be a list of the best orders in the market for the specified contract at each of the specified time intervals, in the following format:

```
[
    {
        "timestamp": 1677671415000000000,
        "bidPrice": 118,
        "bidQuantity": 15,
        "askPrice": null,
        "askQuantity": null
    },
    {
        "timestamp": 1677671430000000000,
        "bidPrice": 118,
        "bidQuantity": 15,
        "askPrice": null,
        "askQuantity": null
    }, […]
]
```

The fields in the response are defined as follows:

- **askPrice**: The price of the best ask in the market at the specified time, quoted in the **currency** value of the instrument or market (see Reference Data API for more information).
- **askQuantity**: The quantity of the best ask in the market, quoted in the **unit** value of the instrument or market.
- **bidPrice**: The price of the best bid in the market, quoted in the **currency** value of the instrument or market.
- **bidQuantity**: The quantity of the best bid in the market, quoted in the **unit** value of the instrument or market.

- **timestamp**: The time of this data point, expressed in nanoseconds since 01/01/1970 for JSON and ISO-8601 format for CSV.

The bid and ask fields describe individual orders or exchange-defined price levels. If there are multiple bids or asks in the market at the same price, the oldest order is described in the quantity fields. For more information please see Level 2 and Level 3 Data.

If there were no orders on one side of the market at a particular point in time (for example, if there were bids present but no asks), the fields for the empty side are set to "null", as is the case for asks in the example above.

If there were no orders on either side of the market at a particular point in time, that time will be omitted from the results unless you have set the **includeEmptyBuckets** parameter to "true", in which case a value of "null" is set in both the bid and ask fields for that time.

If there were no bids or asks in the market for any of the requested points in time, and you did not set the **includeEmptyBuckets** parameter to "true", an empty list ("[]") is returned.

Level 2 (/orders/book)

Returns a timeseries of all bids and asks in the market for the specified contract.

This endpoint accepts the same parameters as the /orders/book/top endpoint described above. Two additional optional parameters can be used to filter the data returned:

- **depth**: The maximum number of bids or asks to be returned on each side of the market. For example, if you set this to "5" then only the best five bids and the best five asks are included in your response.

  If you do not provide a value for this parameter, a default value of "10" is assumed and responses contain only the best 10 bids and the best 10 asks for each point in time. To request the full market depth, set this value to a large number such as "100".
- **maxSpread**: The maximum percentage price difference allowed between the best bid or ask price and the prices of the returned orders. For example, if you set this to "20" then only bids within 20% of the price of the best bid, and asks within 20% of the price of the best ask are returned. If you do not provide a value for this parameter a default of "null" is assumed, and no price restriction is applied to the results.

These two parameters work in combination. If you specify a **depth** of "5" and a **maxSpread** of "20" then you receive the five best-priced bids that are within 20% of the price of the best bid (and likewise for asks).

Similarly, if you omit the **depth** parameter and set **maxSpread** to "20", the default value of "10" for **depth** is assumed and you receive up to the 10 best-priced bids that are within 20% of the price of the best bid (and likewise for asks).

If there are multiple orders at the same price level, and depth restrictions mean that not all of them can be displayed, the oldest orders are preferred.

The /orders/book endpoint also accepts the following optional parameters:

- **includePrivate**: Whether private orders (those that belong to the user or their company) are identified as such. If you do not provide a value for this parameter a default value of "false" is assumed, and only the public (i.e. anonymous, market) versions of orders are returned. When set to "true", private orders will include a **privateFields** object with additional details about the order.
- **routes** as described in Routes to Market.
- **optionalFields**: A list of additional fields to be included in the response. The following additional fields may be requested:
  - **route** and **routeId** as described in Routes to Market.
  - **venueCode**: The Trayport-defined code for the broker or exchange on which the order was placed. A list of these codes can be found in Venue Codes.

    You can specify any number of optional fields, either by separating them by commas in the value of a URL parameter (e.g. "optionalFields=route,venueCode") or by repeating the parameter name (e.g. "optionalFields=route?optionalFields=venueCode").

    If no values are provided for this parameter, only the default trade fields described above are included in the response.

The response is a list of the best bids and asks that meet your **depth** and **maxSpread** criteria for each of the points in time you requested.

For example, to request the best 2 bids and asks for TTF* Apr-23 every 15 seconds between 09:00 and 10:00 UTC on 1st March 2023, including private orders, you could run the following query:

```
https://analytics.trayport.com/api/orders/book?from=2023-03-01T09&until=2023-03-01T10&interval=15&intervalUnit=second&depth=2&includePrivate=true&marketId=10000065&sequenceId=10000305&sequenceItemId=232&contractType=SinglePeriod
```

The response will be in the following format:


```
[
    {
        "timestamp": 1677661215000000000,
        "bids": [
            {
                "price": 47.305,
                "quantity": 10
            },
            {
                "price": 47.255,
                "quantity": 5
            }
        ],
        "asks": [
            {
                "price": 47.35,
                "quantity": 5
            },
            {
                "price": 47.35,
                "quantity": 10,
                "privateFields": {
                    "clientOrderId": "",
                    "orderStatus": "Firm",
                    "lastModifierUserId": 12345,
                    "hiddenQuantity": 0,
                    "ownerUserId": 23456,
                    "ownerCompanyId": 987,
                    "originalHiddenQuantity": 0,
                    "hiddenQtyPriceDelta": 0
                }
            }
        ]
    }, […]
]
```

The fields in the response are defined as follows:

- **timestamp**: The time of this data point, expressed in nanoseconds since 01/01/1970 for JSON and ISO-8601 format for CSV.
- **bids** and **asks**: Lists of the best bids and asks in the market at the specified time. Each order has the following properties:
  - **price**: The price of the order, quoted in the **currency** value of the instrument or market (see Reference Data API for more information).
  - **quantity**: The quantity of the order, quoted in the **unit** value of the instrument or market.
  - If the bid or ask is a private order and you set the **includePrivate** parameter to "true", a **privateFields** object is present on the order in addition to the **price** and **quantity** fields. This object contains the following fields:
    - **clientOrderId**: The Client Order ID set by its owner through the Joule Direct API. Only present if the owner of the order set this ID when creating the order.
    - **orderStatus**: Either "Firm" (active in the market) or "Withheld" (invisible to other market participants and untradable until 'firmed' again).
    - **lastModifierUserId**: The numeric Joule Direct User ID of the last user to make a change to this order.
    - **hiddenQuantity**: The current quantity of this order that is hidden from the market.
    - **ownerUserId**: The numeric Joule Direct User ID of the owner of the order.
    - **ownerCompanyId**: The numeric ID of the company to which the order's owner belongs.
    - **originalHiddenQuantity**: The hidden quantity attached to this order when its volume was last updated, before any of it was dealt.
    - **hiddenQtyPriceDelta**: The price delta associated with the order.

Private fields are also returned for orders belonging to companies over which you have "identify" permissions. More information about this mechanism can be found in the Private Data from Other Companies section.

Level 2 and Level 3 Data (Aggregating by Price Level)

The Data Analytics API does not aggregate orders by price level. If two different orders with the same price are visible on the Joule screen, then two different orders with the same price will be visible in responses from the /orders/book endpoint. However, as public data sent to Trayport by some exchanges is already aggregated by price level, these results therefore include a mixture of "Level 2" (aggregated by price level) and "Level 3" (individual order) data. The following exchanges currently aggregate public orders by price level:

- EEX7
- EEX (Legacy)
- NDAQ
- ICE
- GMEP
- GMEG
- IDEX
- IENX
- MIBG
- NODX.

All other venues provide individual orders instead of aggregated price levels.

Trades (/trades)

Returns a list of trades for the specified contract between the given times.

The following parameters are required:

- Contract specifiers. The /trades endpoint supports querying a single contract or multiple contracts based on a set of sequence items.
  - If querying a single contract, see Contract specifiers as described in Specifying a Contract.
  - If querying multiple contracts, see Contract specifiers as described in Specifying a Set of Sequence Items. If using **tradingItems**, a maximum limit of 50 is supported.
- **from** and **until**: The dates and times between which trade data should be returned. Provide these as ISO-8601 datetimes with no fractional seconds; for example, "2023-03-22T12:34.56Z". Lower precision timestamps are accepted and rounded to the start of the next largest time unit; for example, "2023-03-22" is interpreted as "2023-03-22T00:00:00Z". Similarly, UTC is assumed if you do not specify a time zone.

  The largest time range you can query is 32 days. If you attempt to query a range larger than this you will receive a "400" (bad request) response code.

The following parameters may also be used but are not mandatory:

- **includePrivate**: Whether private trades (those that belong to the user or their company) are identified as such. If you do not provide a value for this parameter a default value of "false" is assumed, and only the public (i.e. anonymous, market) versions of trades are returned. When set to "true", private trades include a **privateFields** object with additional details about the trade.
- **routes** as described in Routes to Market.

- **optionalFields**: A list of additional fields to be included in the response. The following additional fields may be requested:
  - **contract**: Contract specifiers as defined in Specifying a Contract.
  - **route** and **routeId** as described in Routes to Market.
  - **fromBrokenSpread**: Whether the trade is from a spread trade that has been broken.
  - **initiatorOwnedSpread**: Whether the trade occurred as the result of an order being placed on a spread contract and subsequently dealt.
  - **aggressorOwnedSpread**: Whether the trade occurred as the result of a leg order being generated from a spread order, and that leg order being subsequently dealt.
  - **aggressorSleeve**: Whether the trade is a third-party trade where the aggressor was trading on behalf of a company without a trading agreement with the initiator. This field is always "false" on public trades.
  - **initiatorSleeve**: Whether the trade is a third-party trade where the initiator was trading on behalf of the company without a trading agreement with the aggressor. This field is always "false" on public trades.

  You can specify any number of optional fields, either by separating them by commas in the value of a URL parameter (e.g. "optionalFields=fromBrokenSpread,initiatorOwnedSpread") or by repeating the parameter name (e.g. "optionalFields=fromBrokenSpread?optionalFields=initiatorOwnedSpread").

  If no values are provided for this parameter, only the default trade fields described above are included in the response.

For example, to request all TTF* Apr-23 trades between 09:00 and 10:00 UTC on 1st March 2023, including the details of private trades and adding the **fromBrokenSpread** and **initiatorSleeve** optional fields you could run the following query:

```
https://analytics.trayport.com/api/trades?from=2023-03-01T09&until=2023-03-01T10&includePrivate=true&marketId=10000065&sequenceId=10000305&sequenceItemId=232&contractType=SinglePeriod&optionalFields=fromBrokenSpread,initiatorSleeve
```

The response would be in the following format:

```
[
    {
        "tradeId": "12345678",
        "venueCode": "BRO1",
        "dealDate": 1677661222627000000,
        "price": 47.35,
```

```
        "quantity": 5,
        "aggressorBuy": true,
        "fromBrokenSpread": false,
        "initiatorSleeve": false
    },
    {
        "tradeId": "ABC123",
        "venueCode": "BRO2",
        "dealDate": 1677661222627634000,
        "price": 47.35,
        "quantity": 5,
        "aggressorBuy": false,
        "fromBrokenSpread": false,
        "initiatorSleeve": false,
        "privateFields": {
            "aggressorCompanyId": 123,
            "aggressorTraderCompanyName": "ACME Trading Ltd",
            "aggressorTraderId": 4321,
            "aggressorTraderName": "Joe Bloggs",
            "aggressorDerivativeIndicator": true,
            "initiatorCompanyId": 0,
            "initiatorTraderCompanyName": "",
            "initiatorTraderId": 0,
            "initiatorTraderName": "",
            "initiatorDerivativeIndicator": false,
            "productClassification": "RmMifid"
        }
    }, […]
]
```


The fields in the response are defined as follows:

- **tradeId**: The ID assigned to this trade by the venue (broker or exchange) where it was conducted. Some venues generate numeric IDs for their trades but others use alphanumeric strings. We recommend treating this as a 'string' record.

  > **Note:** Each broker or exchange is independently responsible for generating their own trade IDs. This means that it is possible for the same ID to be used by different brokers or exchanges. To create a trade identifier that is unique across all brokers and exchanges, combine the **tradeId** with the **venueCode**.

- **venueCode**: The Trayport-defined code for the broker or exchange where the trade was conducted. A list of these codes can be found in Venue Codes.

- **dealDate**: The time the trade was conducted, as defined by the venue, expressed in nanoseconds since 01/01/1970.
- **price**: The price of the trade, quoted in the **currency** value of the instrument or market (see Reference Data API for more information).
- **quantity**: The quantity of the trade, quoted in the **unit** value of the instrument or market.
- **aggressorBuy**: Whether the aggressor of the order was the buyer.

These six fields are provided for all trades, and are followed by any optional fields you have requested.

If you set the **includePrivate** parameter to "true" and your company was party to a trade, that trade also includes a **privateFields** object containing the names and IDs of the initiator and aggressor traders and companies.

The names and IDs for your side of the trade are as defined in Joule Direct; the names and IDs for your counterparty (if available), are as defined by the broker or exchange on which the trade was conducted. If the identity of your counterparty is not available (e.g. If the trade was conducted on an exchange or was marked by a broker as 'anonymous'), then these fields will be present but with empty or default values ("0" or ""), as shown for the initiator in the example above.

Private fields are also returned for trades belonging to companies over which you have "identify" permissions. More information about this mechanism can be found in the Private Data from Other Companies section.

In addition to these identifiers, the **privateFields** object contains two fields relating to the Mifid II classification of the trade:

- **initiatorDerivativeIndicator**: Whether the initiator of the trade has indicated the trade significantly reduces risk, as defined in the Mifid II Regulatory Technical Standards (RTS) 22 Annex I (Table 2, Field 64).
- **productClassification**: The classification of this trade in relation to Mifid II.

OHLCV (/trades/ohlcv)

Returns the Open, High, Low, Close and Volume ("candlestick") statistics for a defined interval and contract, with the option to add VWAP.

The following parameters are required:

- Contract specifiers as described in Specifying a Contract.
- **from** and **until**: The dates and times between which candlestick data should be returned. Provide these as ISO-8601 datetimes with no fractional seconds; for example, "2023-03-22T12:34.56Z". Lower precision timestamps are accepted and rounded to the start of the next largest time unit; for example, "2023-03-22" is interpreted as "2023-03-22T00:00:00Z". Similarly, UTC is assumed if you do not

specify a time zone.

The time range you can request is limited to 92 days (~3 months) when querying with intervals of less than 1 day, and 1827 days (~5 years) when querying with intervals of at least 1 day. The **from** and **until** values must be aligned with the intervals requested; for example, if querying intervals of 1 day, these values must align with midnight UTC.

- **intervalUnit** and **interval**: Used to define the length of each "candlestick" returned. For example, if you wanted to retrieve 15 minute candles between your given **from** and **until** times, you would set the **intervalUnit** to "minute" and the **interval** to "15". The following values are valid for **intervalUnit**. Only the following combinations of these values can be queried:
  - 1, 5, 15, 30 minutes
  - 1, 4 hours
  - 1, 7 days
  - 1 (calendar) month.

The following optional parameters may also be used:

- **includeEmptyBuckets**: Whether time intervals in which there were no trades are included in the response. If this parameter is not specified it assumes a default value of "false" and periods in which there were no trades are not included in the response. Valid values are "true" or "false".
- **optionalFields**: A list of additional fields to be included in the response. The following additional fields may be requested:
  - **openTimestamp**: The time that the first trade in each interval took place, expressed in nanoseconds since 01/01/1970 for JSON and ISO-8601 format for CSV.
  - **closeTimestamp**: The time that the last trade in each interval took place, expressed in nanoseconds since 01/01/1970 for JSON and ISO-8601 format for CSV. If there was only one trade in a given interval, its values for **openTimestamp** and **closeTimestamp** are the same.
  - **vwap**: The volume-weighted average price of all the trades in the interval. This is calculated by dividing the sum of the product of price and quantity for each trade by the sum the quantity of all trades.

    This is not a cumulative or 'rolling' VWAP. To calculate VWAP from a given point in time, multiply the **vwap** value for each interval from that time by its **volume** to calculate the 'value' of each interval, and then divide the sum of the values by the sum of the **volume** for the same intervals. If you are interested in having this calculation performed automatically by the API please contact your Trayport Client Relationship Manager to request this.

For example, to retrieve the 15-minute "candlesticks" for 27th March 2023 and including the **vwap** value on each for EUA* Dec-23, you could run the following query:

```
https://analytics.trayport.com/api/trades/ohlcv?interval=15&intervalUnit=minute
&optionalFields=vwap&from=2023-03-27&until=2023-03-
28&marketId=10000016&contractType=SinglePeriod&sequenceId=10000400&sequenceItemId=710
```

The response would be in the following format:


```
[
  {
    "fromTimestamp": 1679896800000000000,
    "toTimestamp": 1679897700000000000,
    "open": 88.02,
    "high": 89,
    "low": 88.02,
    "close": 88.22,
    "volume": 305,
    "vwap": 88.5361
  },
  {
    "fromTimestamp": 1679897700000000000,
    "toTimestamp": 1679898600000000000,
    "open": 88.22,
    "high": 88.53,
    "low": 88.22,
    "close": 88.31,
    "volume": 132,
    "vwap": 88.3709
  }, […]
]
```


The fields in the response are defined as follows:

- **fromTimestamp**: The starting time of this interval, expressed in nanoseconds since 01/01/1970 for JSON and ISO-8601 format for CSV.
- **toTimestamp**: The time this interval ends, expressed in nanoseconds since 01/01/1970 for JSON and ISO-8601 format for CSV.
- **open**: The price of the first trade that occurred during the time interval.
- **high**: The price of the trade with the highest price that occurred during the time interval.
- **low**: The price of the trade with the lowest price that occurred during the time interval.
- **close**: The price of the last trade that occurred during the time interval.

- **volume**: The sum of the volume of all the trades that took place during this time interval.

The **open**, **high**, **low** and **close** values are all expressed in the currency of the instrument or market.

The **volume** value is expressed in the unit of the instrument or market.

Last (/trades/last)

Returns the price and time of the last trade that took place on the given contract.

The following parameters are required:

- Contract specifiers as described in Specifying a Contract.

The following optional parameters may also be used:

- **at**: The time at which the last traded price should be calculated. Provide this as an ISO-8601 datetime with no fractional seconds; for example, "2023-03-22T12:34.56Z". Lower precision timestamps are accepted and rounded to the start of the next largest time unit; for example, "2023-03-22" is interpreted as "2023-03-22T00:00:00Z". Similarly, UTC is assumed if you do not specify a time zone.

  If you do not provide a value for the **at** parameter, the latest time available to you is used instead, based on the permissions assigned to your API key:
  - If your key has access to intraday data, the current time is used.
  - If your key does not have access to intraday data, midnight UTC of the current day is used instead.
- **optionalFields**: A list of additional fields to be included in the response. The following additional fields may be requested:
  - **contract**: Contract specifiers as defined in Specifying a Contract.

For example, to retrieve the price of the last TTF* Apr-23 trade from 16:30 on 27th March 2023, you could run this query:

```
https://analytics.trayport.com/api/trades/last?at=2023-03-27T16%3A30&marketId=10000065&contractType=SinglePeriod&sequenceId=10000305&sequenceItemId=232
```

The response would be in the following format:

```
[
  {
    "price": 41.5,
    "dealDate": 1679932931000000000
```

```
  }
]
```

The fields in the response are defined as follows:

- **price**: The price of the last trade that occurred before the specified or default time.
- **dealDate**: The time the trade took place, expressed in nanoseconds since 01/01/1970.

> **Note:**
> • The **at** parameter is exclusive; if a trade occurred exactly at 16:30 in this example it would not be returned and instead the details of the previous trade would be used.
> • Only trades that took place within a year of the given **at** value are considered.

Activity (/trades/activity)

Returns a list of contracts that have traded between the specified times.

The following parameters are required:

- **from** and **until**: The dates and times between which trade activity should be returned. Provide these as ISO-8601 datetimes with no fractional seconds; for example, "2023-03-22T12:34.00Z". Note that the smallest resolution supported by this endpoint is **one minute**: seconds may be specified but will be ignored. Lower precision timestamps are accepted and rounded to the start of the next largest time unit; for example, "2023-03-22" is interpreted as "2023-03-22T00:00:00Z". Similarly, UTC is assumed if you do not specify a time zone.
  - The earliest accepted value for **from** is 10th April 2023. Earlier dates will become available in a future upgrade.
  - The latest accepted value for **until** is 7 days after the **from** value.

The following parameters may also be used but are not mandatory:

- **routes** as described in Routes to Market.

  Note that the **deduplicateRoutes** parameter that is available on /trades and /orders/book is not available in for the Activity endpoint. If multiple routes are requested, a separate contract count will be returned for each route on which trade activity occurred.
- **optionalFields**: A list of additional fields to be included in the response. The following additional fields may be requested:
  - **route** and **routeId** as described in Routes to Market.
  - **marketId**: The numeric ID of the commingled market to which the contract belongs. If the contract does not belong to a market, this will be ‘null’ in a JSON response or empty in a CSV response.

- **updateCount**: The number of times trades on this contract were updated during the requested period.
- **deleteCount**: The number of times trades on this contract were deleted during the requested period.

For example, to retrieve activity on the 1st May 2023 from all available routes, including market ID and the delete count for each contract, you could run the following query:

```
https://analytics.trayport.com/api/trades/activity?from=2023-05-01&until=2023-05-02&routes=all&optionalFields=marketId&optionalFields=deleteCount
```

The response would be in the following format:


```
[
  {
    "instrumentId": 10003007,
    "marketId": 10000016,
    "sequenceId": 10000400,
    "sequenceItemId": 710,
    "secondSequenceItemId": null,
    "contractType": "SinglePeriod",
    "count": 6454,
    "deleteCount": 0,
    "route": "House"
  },
  {
    "instrumentId": 10002228,
    "marketId": 10000065,
    "sequenceId": 10000305,
    "sequenceItemId": 234,
    "secondSequenceItemId": null,
    "contractType": "SinglePeriod",
    "count": 3593,
    "deleteCount": 0,
    "route": "House"
  }, […]
]
```


The fields in each object in the response are defined as follows:

- Contract specifiers as defined below:
  - **instrumentId**: The numeric ID of the instrument corresponding to the contract.
  - **marketId**: The numeric ID of the commingled market to which the contract belongs. If the contract does not belong to a market, this field will be 'null' in a JSON response or empty in a CSV response.
  - **sequenceId**: The numeric ID of the sequence containing the traded delivery period of the contract.
  - **sequenceItemId**: The numeric ID of the traded delivery period of the contract (aka "sequence item").
  - **contractType**: Whether the contract has multiple delivery periods, and if so, of what type. The following values may be used:
    - **SinglePeriod**: Used if the contract has a single delivery period (e.g. Apr-23).
    - **PeriodSpread**: Used if the contract has a period (aka "time" or "calendar") spread (e.g. Apr-23 x May-23).
    - **PeriodRange**: Used if the contract is a period range (e.g. for all periods between Apr-23 and Aug-23).
    - **secondSequenceItemId**: If the contract is a period spread or range, the numeric ID of the second sequence item in the spread or range. For single periods, this field is 'null' in a JSON response or empty in a CSV response.
- **count**: The number of times trades were inserted on this contract during the requested period.

  Note that if trades are subsequently deleted, this number will not be reduced; instead, the value in the **deleteCount** optional field will be increased. This includes deleted trades that are no longer visible in the results of queries to the /trades endpoint (see Deleted Trades for more information).

Private Trades (/trades/private)

Returns a list of trades belonging to companies over which you have "identify" permissions (your own company's trades, along with those from subsidiaries, partners or DMA clients) between the specified times. Public (aka 'anonymous' or 'market') trades are not included in the response as they are on the /trades endpoint.

The following parameters are required:

- **from** and **until**: The dates and times between which trade data should be returned. Provide these as ISO-8601 datetimes with no fractional seconds; for example, "2023-03-22T12:34.56Z". Lower precision timestamps are accepted and rounded to the start of the next largest time unit; for example, "2023-03-22" is interpreted as "2023-03-22T00:00:00Z". Similarly, UTC is assumed if you do not specify a time zone.

  The largest time range you can query is 32 days. If you attempt to query a range larger than this you will receive a "400" (bad request) response code.

> **Note:** Private trades may be retrieved from the current day without the need for the Data Analytics Gateway or any Intraday permissions.

The following parameters may also be used but are not mandatory:

- Contract specifiers. If not provided, trades for all contracts are returned. The /trades/private endpoint supports querying a single contract or multiple contracts based on instrument, market or sequence.
  - If querying a single contract, see Contract specifiers as described in Specifying a Contract.
  - If querying multiple contracts, see Contract specifiers as described in Specifying All Contracts for a Given Market, Instrument or Sequence.
- All other optional parameters from the /trades endpoint may also be used, with the exception of **includePrivate**.

For example, to request all private trades from 1st and 2nd of June from all routes to market and adding the **fromBrokenSpread** optional field, you could run the following query:

```
https://analytics.trayport.com/api/trades/private?from=2023-06-01&until=2023-06-03&optionalFields=fromBrokenSpread&routes=all&deduplicateRoutes=true
```

The response would be in the following format:


```
[
  {
    "tradeId": "12345678",
    "venueCode": "BRO1",
    "dealDate": 1685600903935000000,
    "lastUpdated": 1685600903935000000,
    "price": 25.375,
    "quantity": 100,
    "aggressorBuy": true,
    "fromBrokenSpread": false,
    "route": "House",
    "aggressorCompanyId": 0,
    "aggressorTraderCompanyName": "",
    "aggressorTraderId": 0,
    "aggressorTraderName": "",
    "aggressorDerivativeIndicator": false,
    "initiatorCompanyId": 123,
    "initiatorTraderCompanyName": "ACME Trading Ltd",
```

```
    "initiatorTraderId": 98765,
    "initiatorTraderName": "Joseph Bloggs",
    "initiatorDerivativeIndicator": false,
    "contract": {
      "contractType": "SinglePeriod",
      "marketId": 10000065,
      "instrumentId": 10002806,
      "sequenceId": 10000302,
      "sequenceItemId": 2,
      "secondSequenceItemId": null
    }
  }, […]
]
```


The fields in the response match those in the response to /trades queries.

If you did not fully specify a contract in the request, a **contract** child object is added to each trade defining the contract for which it was made. The fields in this object are defined below:

- **contractType**: Whether the contract has multiple delivery periods, and if so, of what type. The following values may be used:
  - **SinglePeriod**: Used if the contract has a single delivery period (e.g. Apr-23).
  - **PeriodSpread**: Used if the contract has a period (aka "time" or "calendar") spread (e.g. Apr-23 x May-23).
  - **PeriodRange**: Used if the contract is a period range (e.g. for all periods between Apr-23 and Aug-23).
- **instrumentId**: The numeric ID of the instrument corresponding to the contract.
- **marketId**: The numeric ID of the commingled market to which the contract belongs. If the contract does not belong to a market, this field is 'null' in a JSON response or empty in a CSV response.
- **sequenceId**: The numeric ID of the sequence containing the traded delivery period of the contract.
- **sequenceItemId**: The numeric ID of the traded delivery period of the contract (aka "sequence item").
- **secondSequenceItemId**: If the contract is a period spread or range, the numeric ID of the second sequence item in the spread or range. For single periods, this field is 'null' in a JSON response or empty in a CSV response.

Snapshots (/snapshots)

Returns the price and quantity of the best bid and ask at a particular point in time, as well as the time and price of the last trade for a given set of contracts.

The following parameters are required:

- Contract specifiers. The /snapshots endpoint supports querying a single contract or multiple contracts based on a set of sequence items.
  - If querying a single contract, see Contract specifiers as described in Specifying a Contract.
  - If querying a set of contracts, see Contract specifiers as described in Specifying a Set of Sequence Items. If using **tradingItems**, a maximum limit of 50 is supported.

A single optional parameter may also be used:

- **at**: The time at which the best bid and ask orders and last traded price should be returned. Note that the smallest resolution supported by this endpoint is **one minute**. Provide this as an ISO-8601 datetime with no fractional seconds; for example, "2023-03-22T12:34.00Z". Lower precision timestamps are accepted and rounded to the start of the next largest time unit; for example, "2023-03-22" is interpreted as "2023-03-22T00:00:00Z". Similarly, UTC is assumed if you do not specify a time zone.

  If you do not provide a value for the **at** parameter, the latest time available to you is used instead, based on the permissions assigned to your API key:
  - If your key has access to intraday data, the current time is used.
  - If your key does not have access to intraday data, midnight UTC of the current day is used instead.

For example, to retrieve the snapshot of TTF* Gas Months for 2 trading items where the current date is 15/08/2023, you could run this query:

```
https://analytics.trayport.com/api/snapshots?marketId=10000065&sequenceId=10000305&contract
Type=SinglePeriod&tradingItems=2
```

The response would be in the following format:

```
"timestamp": 1692113400000000000,
"snapshots": [
    {
        "contract": {
            "contractType": "SinglePeriod",
```

```
            "marketId": 10000065,
            "instrumentId": null,
            "sequenceId": 10000305,
            "sequenceItemId": 237,
            "secondSequenceItemId": null
        },
        "lastTrade": {
            "price": 39.25,
            "dealDate": 1692113368492000000
        },
        "bestOrders": {
            "bidPrice": 38.88,
            "bidQuantity": 1,
            "askPrice": 39.1,
            "askQuantity": 1
        }
    },
    {
        "contract": {
            "contractType": "SinglePeriod",
            "marketId": 10000065,
            "instrumentId": null,
            "sequenceId": 10000305,
            "sequenceItemId": 238,
            "secondSequenceItemId": null
        },
        "lastTrade": {
            "price": 43.905,
            "dealDate": 1692113093180583338
        },
        "bestOrders": {
            "bidPrice": 43.96,
            "bidQuantity": 3,
            "askPrice": 44.175,
            "askQuantity": 3
        }
    }
]
```


The fields in the response are defined as follows:

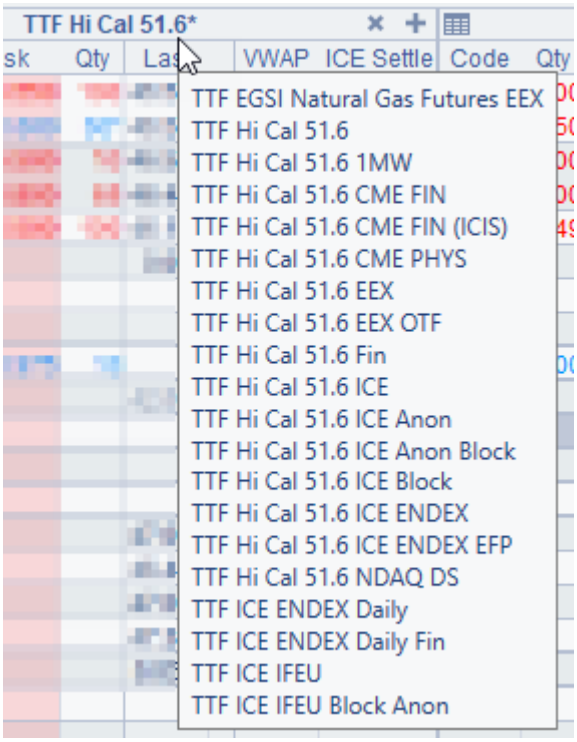
- **timestamp**: The time of snapshot order data, expressed in nanoseconds since 01/01/1970 for JSON and ISO-8601 format for CSV.
- **snapshots**: List of **contracts** with a **lastTrade** object and **bestOrders** object.
- **contract**: Contract specifiers as defined below:
  - **contractType**: Whether the contract has multiple delivery periods, and if so, of what type. The following values may be used:
    - **SinglePeriod**: Used if the contract has a single delivery period (e.g. Apr-23).
    - **PeriodSpread**: Used if the contract has a period (aka "time" or "calendar") spread (e.g. Apr-23 x May-23).
    - **PeriodRange**: Used if the contract is a period range (e.g. for all periods between Apr-23 and Aug-23).
  - **instrumentId**: The numeric ID of the instrument corresponding to the contract.
  - **marketId**: The numeric ID of the commingled market to which the contract belongs. If the contract does not belong to a market, this field is 'null' in a JSON response or empty in a CSV response.
  - **sequenceId**: The numeric ID of the sequence containing the traded delivery period of the contract.
  - **sequenceItemId**: The numeric ID of the traded delivery period of the contract (aka "sequence item").
  - **secondSequenceItemId**: If the contract is a period spread or range, the numeric ID of the second sequence item in the spread or range. For single periods, this field is 'null' in a JSON response or empty in a CSV response.
- **lastTrade** object with the following properties:
  - **price**: The price of the last trade that occurred before the specified or default time.
  - **dealDate**: The time the trade took place, expressed in nanoseconds since 01/01/1970 for JSON and ISO-8601 format for CSV.
- **bestOrders** object with the following properties:
  - **bidprice**: The price of the best bid order.
  - **bidquantity**: The quantity of the best bid order.
  - **askprice**: The price of the best ask order.
  - **askquantity**: The quantity of the best ask order.

Note:
• The **at** parameter is exclusive; if a trade occurred exactly at 16:30, in this example it would not be returned and instead the details of the previous trade would be used.
• Only trades that took place within a year of the given **at** value are considered.
• Only orders that are in the market at the specified time are returned.

SPECIFYING A CONTRACT

Most queries in the Analytics API are issued against a specific contract. In Trayport's data model, a contract is a combination of a product and a delivery period (for example, "NBP Apr-23"). Delivery periods are grouped into Sequences of similar maturities. For example, there are sequences for Gas Months, Power Quarters, and Coal Years.

There are two kinds of product you can query in the Analytics API: individual instruments such as "TTF Hi Cal 51.6 EEX", and commingled markets (also known as "All Venues" products in Joule). A commingled market comprises a number of related instruments that Trayport has grouped for ease of comparison. For example, the "TTF Hi Cal 51.6*" commingled market contains physical and financial TTF instruments from multiple brokers and exchanges.



To specify a contract, you must provide values for the following parameters:

- **instrumentId**: The numeric ID of an instrument. If you use this parameter, you may not also specify a **marketId**. For example, to query "TTF Hi Cal 51.6 EEX" you can use the numeric ID "10002806".
- **marketId**: The numeric ID of a commingled market. If you use this parameter, you may not also specify an **instrumentId**. For example, to query the "TTF Hi Cal 51.6*"" All Venues market, you can use the numeric ID "10000065".
- **sequenceId**: The numeric ID of the sequence containing the delivery period you wish to query. For example, to query for Apr-23 for TTF you would need to specify the ID of the "Gas - NG Months" sequence ("10000305").

- **sequenceItemId**: The numeric ID of the delivery period (sequence item) you wish to query. For example, Apr-23 in the "Gas - NG Months" sequence has the ID "232".
- **contractType**: Whether the contract has multiple delivery periods, and if so, what type. The following values are permitted:
  - **SinglePeriod**: Used if the contract has a single delivery period (e.g. Apr-23).
  - **Spread**: Used if the contract has a period (time or calendar) spread (e.g. Apr-23 x May-23).
  - **Range**: Used if the contract is a period range (e.g. for all periods between Apr-23 and Aug-23). Ranges are not commonly used outside intraday power trading.
- **secondSequenceItemId**: Only required if you are requesting a **Spread** or **Range**. In this case, the ID of the second sequence item should be specified. If you are requesting data for a **SinglePeriod** contract, this field should be omitted.

The names and numeric IDs of instruments, sequences and items are the same in all Trayport applications, including Joule and the Joule Direct API. To assist you with determining the IDs to use, the Reference Data API is supplied alongside the Analytics API (see Reference Data API for more information).

SPECIFYING MULTIPLE CONTRACTS

SPECIFYING ALL CONTRACTS FOR A GIVEN MARKET, INSTRUMENT OR SEQUENCE

The trades/private endpoint supports querying all contracts for a given market, instrument or sequence.

To query all contracts for a specific market or instrument, you must provide values for the following parameters:

- **marketId** or **instrumentId**

To query all contracts for a specific sequence you must provide values for the following parameters:

- **sequenceId**

and optionally:

- **marketId** or **instrumentId**.

SPECIFYING A SET OF SEQUENCE ITEMS

The /trades and /snapshots endpoints support querying multiple contracts from a set of sequence item IDs. These can be specified either as separate **sequenceItemId** parameters, or as a single parameter with values separated by commas. This is only supported where **contractType** is **SinglePeriod**.

To specify a set of contracts using multiple sequence items, you must provide values for the following parameters:

- **instrumentId** or **marketId**
- **sequenceId**
- **sequenceItemId**: As separate parameters or a comma separated list. For example, Apr-23, May-23 and Jun-23 in the "Gas - NG Months", the list of sequence IDs is "232,233,234".
- **contractType**: The following values are permitted:
  - **SinglePeriod**

Alternatively, instead of specifying a list of specific sequence items, the **tradingItems** parameter can be used. Using this parameter will result in a query for all contracts for the specified sequence starting with the first tradable sequence item up to the final tradable item within the time period of the query. The **tradingItems** parameter has a limit to the items that can be queried; this limit is defined per endpoint where it is supported (please see the documentation for the relevant endpoint). The **tradingItems** parameter cannot be used together with **sequenceItemId**.

The **tradingItems** parameter is only supported:

- where **contractType** is **SinglePeriod**
- for sequences where the sequence items have a defined trading end date. This excludes certain sequences such as rolling gas prompt contracts and emissions contracts. A sequence item has a defined trading end date if such a date is returned on the /sequences/{sequenceId}/sequenceItems query in the Reference Data API.

To specify a set of contracts using multiple sequence items based on **tradingItems**, you must provide values for the following parameters:

- **instrumentId** or **marketId**
- **sequenceId**
- **tradingItems**
- **contractType**: The following values are permitted:
  - **SinglePeriod**

EXAMPLE

- Query time range is 10th August 2023 to 12th August 2023
- **instrumentId**: 10001126 (Germany Baseload)
- **sequenceId**: 10000105 (Euro - Euro Quarters)
- **contractType**: **SinglePeriod**
- **tradingItems**: 50 (which is also the limit for the example endpoint).

The result is 45 contracts are queried, starting from the first tradable item (Q423) to the final tradable item (Q434).

ROUTES TO MARKET

If your company has access to multiple routes to market – for example, through a DMA provider – you can request data from one or more routes using the following optional parameters:

- **routes**: A list of the route names or IDs you wish to query, either as separate parameters, or in a single parameter separated by commas.
  - You can use the keyword "all" to receive data from all available routes without having to individually list each one.
  - If you do not specify a value for this parameter, only data from the route with the lowest numeric ID (normally your *House* route) is returned.
- **deduplicateRoutes**: Whether trade or order data from multiple routes should be deduplicated. If you have a connection to a broker or exchange on multiple routes, you are likely to receive the same order or trade data from each route. If this parameter is set to "true", the data is deduplicated according to the following rules:
  - If an order or trade is private, and you have set the **includePrivate** parameter to "true", then the private version of that order or trade is used.
  - If an order is tradeable through one route and not through another, the tradeable version of the order is used.
  - For all public trades, and for public orders of the same tradability, trades and orders from the highest-priority route are used instead of orders from the lower-priority routes.

  This logic is equivalent to that used when deduplicating data from multiple routes in the Joule screen.

  If you do not provide a value for this parameter and you have requested data from multiple routes, it is set to "true" by default and data is deduplicated. If you set this value to "false", you will receive a copy of each order and trade from each route to market through which it has been sent to you.

These parameters are available for the following endpoints:

- /trades
- /orders/book
- /orders/book/top

These endpoints also support two optional fields related to routes to market:

- **route**: The name of the route through which the order or trade has been sent.
- **routeId**: The numeric ID of the route through which the order or trade has been sent.

These fields can be requested by adding their names to the **optionalFields** parameter in the usual way. If you have provided any values in the **routes** parameter, the **route** optional field is automatically included in your results and you do not need to request it.

AVAILABLE DATA

SUPPORTED VENUES

Order and trade information is available for all brokers and exchanges to which your company is connected through Joule Direct, apart from CME.

If you would like to access order and trade history from CME, please contact your Trayport Client Relationship Manager.

INTRADAY DATA

By default, the Analytics API returns records up until the start of the current day in Universal Coordinated Time (UTC). Queries with an end time after the start of the current data will not return a successful result. Querying for data after this point requires the Data Analytics Gateway to be installed locally. For more information on accessing Intraday data, please see the Data Analytics Gateway documentation.

HISTORICAL DATA

The age of the data available through the Analytics API varies by type and source, and where applicable, is subject to your relevant historical data agreement with the venue.

Trades

All public and private trades you have received through Trayport from supported venues are available, with no age restrictions.

Orders

All public orders you have received through Trayport from supported broker venues are available from the end of January 2020.

For exchange public orders, the following applies if you have a connection to the respective exchange, regardless of when your company started using Trayport:

- EEX, ICE and IENX: public orders are available from February 2020
- EPEX, GMEG, GMEP, IDEX, MIBG, NDAQ and NODX: public orders are available from February 2021.

Broker and exchange private orders (i.e. those belonging to companies over which the user has "identify" permissions) are available from February 2021.

> **Note:** These dates are not relative dates - Trayport anticipates always being able to provide data starting from the dates given above.

SUPPORT FOR SPREADS

The Analytics API supports the use of both calendar spreads (also called period spreads or time spreads) as well as product spreads (also called location spreads). For order book queries, only firm and venue implied prices are used to build the historic view. At present, implied prices are not available using the Analytics API.

PRIVATE DATA FROM OTHER COMPANIES

If your company has *identify* permissions over the activity of other companies (for example, subsidiaries, partners or DMA clients), all the private order and trade information that would be visible in Joule and the Joule Direct API is also available using the Analytics API.

DELETED TRADES

Trades that have been deleted by the broker or exchange with whom they were conducted are not available through the Analytics API. This includes trades on product or period spreads that have been *broken* (i.e. deleted and new trades on the underlying legs inserted).

COMPARISONS WITH THE JOULE DIRECT API

For supported venues, all orders and trades that are visible through the Joule Direct API are also available in the Analytics API, with the following exceptions:

- Native implied prices (i.e. those calculated by the Joule Direct API or Joule) are not available in the Analytics API.
  - Venue-implied prices (i.e. those calculated by the broker or exchange) are included in the Analytics API.
- Only the visible quantities of hidden-quantity and Iceberg orders are available in the Analytics API.
- Ghost orders are not visible in the Analytics API.

Not all orders and trade properties that are visible in the Joule Direct API are available in the Analytics API. For a full comparison we recommend reviewing the properties exposed on the Swagger pages described in Technical Overview. If you need access to particular fields that are not currently available in the Analytics API, please contact your Trayport Client Relationship Manager.

PUBLIC AND PRIVATE DATA

Brokers provide clients with a single data feed that contains both private and public order and trade information, as do some exchanges. However, most exchanges separate public and private data into two feeds. Consequently, private order and trade actions from these exchanges are sent to Trayport twice; once as a private action and once as an anonymous public action. The Analytics API displays these as two separate order or trade records, in the same way as the Joule Direct API and the MAR End-of-Day Activity File.

For exchanges that provide public and private data separately:

- Queries made to the **/orders/book** or **/trades** endpoints with the *includePrivate* parameter set to *true* return private orders and trades twice – once as private orders or trades, with their private attributes set, and once as anonymous public orders or trades.

For all other exchanges and all brokers:

- Queries made to the **/orders/book** or **/trades** endpoints with the *includePrivate* parameter set to *true* return a single result for each private order or trade, with its private attributes set.
- Queries made to the **/orders/book** or **/trades** endpoints with the *includePrivate* parameter set to *false* return a single result for each private order or trade, without any private attributes set (and thus appearing as though they are public orders and trades).

The following exchanges provide public and private data separately:

- EEX7
- EEX (Legacy)
- NDAQ
- ICE
- GMEP
- GMEG
- IDEX
- IENX
- MIBG
- EPEX
- NODX.

All other venues provide combined public and private data.

REFERENCE DATA API

Most order and trade queries in the Analytics API require an individual contract to be specified. A 'contract' is defined as a combination of:

- An Instrument or commingled Market ID
- A Sequence ID
- A Sequence Item ID
- A second Sequence Item ID (for time spreads or ranges)
- The contract type (single, spread or range)

These IDs can be retrieved from the Reference Data API, which is an additional REST API that accepts the same API key as the Analytics API.

> **Note:** The instrument, sequence and sequence item IDs used in the Analytics and Reference Data API match those used in the Joule Direct API. Market IDs are only available from the Reference Data API and are not used in the Joule Direct API.

The base URL for the Reference Data API is: https://referencedata.trayport.com

The following endpoints are available:

/instruments

Returns a list of all the instruments that can be traded through Trayport's systems. For each instrument, the name and numeric ID will be given.

/markets

Returns a list of all commingled markets (also known as 'All Venues' products). For each market, the name and numeric ID will be given.

/instruments/{instrumentId}

/markets/{marketId}

Returns the details of the market or instrument with the given ID. The following properties are available:

- **Price Currency**: The currency in which prices are quoted.
- **Quantity Unit**: The unit in which the instrument or market is traded.
- **Notional Quantity Unit**: The unit in which prices are quoted and in which the notional quantity of an order or trade is expressed.
- **Notional Currency**: The currency used to express the notional value of an order or trade.

For example, to retrieve the properties of the TTF Hi Cal 51.6 instrument, which has numeric ID 10002096, you would issue a GET request to the following URL: https://referencedata.trayport.com/instruments/10002096

This would return:


```
{
   "id": 10002096,
   "name": "TTF Hi Cal 51.6",
   "tradingSpecifications": {
      "priceCurrency": "EUR",
      "quantityUnit": "MW",
      "notionalCurrency": "EUR",
      "notionalQuantityUnit": "MWh"
   }
}
```


This means that prices for this instrument are quoted in €/MWh, with order and trade quantities listed in MW. The notional volume of a trade would be calculated in MWh and priced in €.

/instruments/{instrumentId}/sequences

/markets/{marketId}/sequences

Returns a list of the sequences that can be used for the given instrument or market. For each sequence, the name, numeric ID, and display name (if applicable) is returned.

/sequences/{sequenceId}/sequenceItems

Returns the sequence items for the sequence with the given ID. For each sequence item, the name and numeric ID is given, along with the start and end dates of the delivery and trading periods where available.

The following parameters may also be used but are not mandatory:

- **count**: The maximum number of sequence items returned. If a value for this parameter is not provided, a default value of "100" is assumed.
- **startDate**: The earliest period end date of a returned sequence item. If a date is not specified, the current date is used. A time and time zone may be specified, but will be ignored.

/sequences/{sequenceId}/sequenceItems/{sequenceItemId}

Returns the sequence item for the given combination of sequence and sequence item IDs. For the sequence item, the name and numeric ID is given, along with the start and end dates of the delivery and trading periods where available.

TECHNICAL OVERVIEW

The Analytics API is documented using the Open API documentation format and can be viewed with the online Swagger UI (see URL Endpoints below for more information) and imported into tools such as postman. The Open API documentation contains the full definition for all request and response types and will be updated when functionality is made available through the API.

URL ENDPOINTS

Analytics API

Root URL: https://analytics.trayport.com/

Documentation: https://www.trayport.com/en/support/daapi/index.html

Reference Data API

Root URL: https://referencedata.trayport.com/

Documentation: https://www.trayport.com/en/support/refdataapi/index.html

MAKING A REQUEST

Requests to the API are performed by calling the HTTP endpoint using the format given in the API Swagger Documentation.

SPECIFYING PARAMETERS

The Analytics API only processes parameters that are supported for the endpoint in question, as defined in the corresponding section of this guide and the API Swagger Documentation. If any unsupported parameters are specified, these are ignored.

AUTHORISATION

In order to authorise a request, an API key must be specified. API keys are generated by Trayport Support and are associated with a user.

When making a request, the API key is passed in on every request to the API service in the 'X-API-KEY' header.

SUPPORTED RESPONSE TYPES

The Analytics API supports JSON and CSV responses. To select the response type, use the *Accept* header with 'application/json' for responses in JSON, or 'text/csv' for responses as CSV format.

The format of each response can be found in the technical swagger documentation.

RATE LIMITS

Requests made against the Analytics API are limited to **eight requests per second** or **480 requests per minute**. These limits are applied separately to each API key.

COMPRESSION

The Analytics API supports compressed responses. To change the compression type, use the standard *Accept-Encoding* header with one of the following values:

- gzip
- deflate
- br

The default value is Brotli (br).

VENUE CODES

The table below provides the codes, names and numeric IDs of all the venues currently available through the Data Analytics API.

> **Note:** As the Data Analytics API can return historical data, it is possible to retrieve orders and trades for venues that are no longer active. The current status of each venue is indicated in the *Active* column.

Code	Name	ID	Active
42FS	42 Financial	8	TRUE
ACGM	Arraco Global Markets	54	FALSE
ACX	Air Carbon	2175	TRUE
AGR	Agrodity	2172	TRUE
ATLA	Braemar Securities Ltd	9	TRUE
BALK	Balkaner	87	FALSE
BETP	BETP	1593	TRUE
BGC	BGC	10	TRUE
BGH	BGH	1028	TRUE
BPIF	Bright Point International Financial (SG) Pte Ltd	1432	TRUE
BTON	Brassington FFA	86	FALSE
CEGE	CEGH	16	FALSE
CIMD	CIMD	13	TRUE
CLKS	Clarksons	21	TRUE
EBI	Energy Brokers Ireland	51	FALSE
EEB	Eagle Energy Brokers	1392	FALSE
EEX	EEX (Legacy)	27	FALSE
EEX7	EEX7	1441	TRUE
EEXS	EEX Spot	20	TRUE
ENEC	enechain Corporation	1298	TRUE
EPEX	EPEX Spot	29	TRUE
EVOL	Evolution Markets	11	TRUE
FINA	Finacor	1565	TRUE
FIS	Freight Investor Services Pte Ltd	774	TRUE
GFI	GFI	2	TRUE
GFIO	GFI Oil	1052	TRUE

Code	Name	ID	Active
GLOB	GlobalCoal	12	TRUE
GMEG	GME Gas	480	TRUE
GMEP	GME Power	134	TRUE
GNGA	Ginga	22	FALSE
GORE	GlobalOre Broker	23	FALSE
GRFN	Griffin Markets Europe SAS	3	TRUE
HXCX	HUDEX/CEEGEX	17	TRUE
IBEX	Independent Bulgarian Energy Exchange	460	TRUE
IBGH	Iberian Gas Hub	34	FALSE
ICAP	ICAP	4	TRUE
ICE	Intercontinental Exchange	30	TRUE
IDEX	IDEX	135	FALSE
IENX	ICE ENDEX Spot	148	TRUE
LNKE	Link Brokerage Enerji	142	FALSE
MAKE	MAKE Brokerage	52	TRUE
MCQL	McQuilling Brokerage Partners	265	FALSE
MIBG	MIBGAS S.A.	827	TRUE
MXFR	Marex Freight	14	TRUE
MYST	MySteel Singapore	75	TRUE
NDAQ	NASDAQ	28	TRUE
NODX	Nodal	946	TRUE
OILB	Oil Brokerage	89	TRUE
OMIP	OMIP	19	TRUE
OPCO	OPCOM	25	TRUE
OTCX	OTCex	15	TRUE
PFL	PFL Futures Limited	879	FALSE
PREB	Tullett Prebon	6	TRUE
PVM	PVM	35	TRUE
PXE	PXE	32	FALSE
SHRD	Shard	47	FALSE
SKM	Svensk Kraftmakling AB	331	FALSE
SPEC	Marex Spectron	5	TRUE
STAR	Starfuels	26	FALSE

Code	Name	ID	Active
TFS	Tradition	7	TRUE
TFSE	Tradition Singapore Pte Ltd	707	TRUE
TSEC	Tavira	123	FALSE
VELT	Velocity Trade Alternative Products	1874	TRUE
VGM	Vanir Global Markets	1950	TRUE
VIKI	Viking Investor Services	1048	FALSE
ZARK	Zark Capital	186	FALSE

FUTURE ENHANCEMENTS

We will continue to bring improvements to both the Trayport Data Analytics platform and the Analytics API. As such, we welcome feedback regarding the current functionality and suggestions for any further functionality we could add in the future. Please contact your Trayport Client Relationship Manager or Trayport Support for further information.