

EEX Market Data – API
User Guide

Lucas Villanueva
27.01.2026
Leipzig

Ref. 21

Table of Contents

1.	Versions	4
2.	Glossary	5
3.	Introduction	6
3.1	Scope	6
3.2	Overview	6
3.3	Subscription	6
4.	Included Market Data	7
5.	Market Data API	8
5.1	Request Method	8
5.2	Symbols	9
5.3	EndPoint	11
5.4	Authentication	12
5.5	Definition of API Formats	12
5.6	Definition of API Data Fields	13
5.8	Market Data API functions, available fields, usage and return format	16
5.8.1	Principles	16
5.8.2	getPermissions	16
5.8.3	getProduct	17
5.8.4	getDerivatives	19
5.8.5	getSpot	23
5.8.6	getAuction	27

5.8.7	getIndex	29
5.8.8	getHistory	37
5.8.9	getQuote	39
5.8.10	getIntraday	41
6.	API Samples for Java and C#	43
6.1	Java Example Request	43
6.2	C# Example Request	45
6.3	API Responses	46
6.3.1	JSON sample return	46
6.3.2	XML sample return	47
6.3.3	CSV sample return	47
7.	Good to Know	48
7.1	Agricultural Indices	48
7.2	Directly addressing Daily and Monthly EGIX via getHistory	48
7.3	Market area CZ VTP / OTE	48
7.4	New contracts first availability	48
7.5	No Gas Spot Data in getIntraday	49
7.6	DeliveryStart/End and TradingStart/End for Derivatives are wrong	49
7.7	Unit of Volumes for Japanese Power Futures	49
7.8	Difference of “old and new” EGSI and EGSI Futures	49
7.9	Uniqueness of trade ID	50
7.10	Best bid best ask availability	50
7.11	Emission Spot availability in getHistory	51
7.12	No Trade Registrations in getIntraday	51
8.	Appendices	52

1. Versions

Version	Description	Date	Author
001	Initial Draft	02.02.2019	Simon Jackson
002	Review	19.03.2019	Daniel Köhler, Mathias Ponnwitz
003	Review	03.04.2019	Simon Jackson, Daniel Köhler, Mathias Ponnwitz
004	Clarify & added new fields after the first few tests	22.07.2019	Daniel Köhler
005	Review and format	23.07.2019	Simon Jackson
006	Review	26.07.2019	Daniel Köhler
007	Review and format	02.09.2019	Simon Jackson
008	Review	03.09.2019	Daniel Köhler
009	Added Best Bid, Best Ask to getQuote & getIntraday. Added DeliveryDate as parameter for getIndex for APDD index. Updated examples to reflect the current syntax.	14.11.2019	Daniel Köhler
010	Review	03.06.2021	Daniel Köhler
011	Additional clarification on the return of Best Bid, Best Ask information for getQuote and getIntraday calls.	28.06.2021	Daniel Köhler
012	Addition of symbol description and additional sample requests	12.07.2022	Jana Credé
013	Additional examples, changes in getHistory and getPermission	19.04.2023	Jana Credé
014	Addition of HYDRIX, EGSI and good to know section	16.10.2023	Jana Credé
015	Additions to good to know section and glossary	16.02.2024	Jana Credé
016	Addition to good to know section and WDRP	11.07.2024	Jana Credé
017	Addition to good to know section and GOs	03.09.2024	Jana Credé
018	Clarified 'OpenPrice' in 5.6 <i>Definition of API Data Fields</i> in regards that OTC trades are not considered.	14.10.2024	Daniel Köhler
019	Clarified request for EGSI index in 5.8.6 <i>getIndex</i>	17.01.2025	Daniel Köhler
020	Introduction of principles for API use in 5.8.1, added data coverage period in 5. Clarification on the field DeliveryPeriod in 5.6.	19.08.2025	Miguel Steinert
021	Clarification in section 7.9 on implicit trades: the Price field is empty, and the impact on OHLL vs. volume/trade count data is detailed.	26.01.2026	Lucas Villanueva

2. Glossary

Term	Description
A	
API	Application Programming Interface
B	
Base load	Base load refers to the load profile of power deliveries with a constant output over 24 hours of every day of the delivery period.
C	
CSV	Comma-separated values
E	
EEX	European Energy Exchange
Endpoint	A URL address which services the API requests
EUA	EU Allowance - One EUA confers the right to emit one tonne of CO ₂ equivalent. An EUA is the smallest tradable unit in EU emissions trading.
EUAA	EU Aviation Allowance. Special EU emission allowances which can only be used by airline companies for compliance purposes.
J	
JSON	JavaScript Object Notation
M	
MDS	Market Data Services
Method	An operation that takes parameters to create a specific request for data
O	
Open Interest	The Open Interest refers to the total of all derivatives contracts which have been opened (i.e. not yet settled) at a given point in time. The Open Interest published by EEX considers all opened positions regardless if a position has been closed with a counter trade. The Open interest provided is not netted.
P	
Peak load	This refers to a load profile for power deliveries with a constant output over twelve hours from 8 a.m. to 8 p.m. on any business day of the delivery period.
S	
Settlement price	Daily market price of a future or option contract which is established by the exchange and used for its daily settlement.
T	
Trade	The API includes both exchange trades and trade registrations (OTC). Implicit spreads are displayed as two separate trades with the same trade ID.
Trade registration	A service which enables the trading participants to register transactions concluded over the counter (OTC transactions) on the exchange, as a result of which clearing and settlement are provided by the clearing house.
U	
Underlying	The subject matter of a futures or option contract is referred to as the "underlying asset". The underlying can be the delivery of a commodity (AG, power, natural gas, emissions, GO, hydrogen), but also an index (e.g. Phelix) or - in the case of options - a future.
URL	Uniform Resource Locator
UTC	Universal Time Coordinated
X	
XML	eXtensible Markup Language

3. Introduction

3.1 Scope

The scope of this document is to provide a programmer who is familiar with web service Application programming interface (web API) with the information required to retrieve market data. It is also intended for non-programmer users to gain a basic understanding on how our API works and how it can be navigated to access the desired data.

3.2 Overview

The API User Guide provides information about how to retrieve EEX market data. This service provides real time updates of the market data available from EEX. The web service API returns XML and JSON exports. To access all EEX data products, the user has to use specific credentials. After purchasing a subscription in our [webshop](#), the user will receive the EEX ID and a password via mail.

An API is defined as a set of functions and procedures allowing the creation of applications that access the features or data of an operating system, application, or other service.

3.3 Subscription

The API solution allows users to receive intraday and end of day EEX market data as well retrieve historical data. Get connected to EEX Group DataSource with fast and robust data directly from the source. Please find all information regarding ordering, prices and available modules of the API on our website.

How to subscribe:

<https://www.eex.com/en/market-data/eex-group-datasource/api>

Contact:

Information Services
T +49 341 2156-288
datasource@eex-group.com

4. Included Market Data

EEX Group DataSource offers the following APIs for market data:

- EEX Power Derivatives
- EEX Natural Gas (Spot and Derivatives)
- EEX Environmental Markets (Spot and Derivatives)
- EEX European Renewable Power GOs
- EEX Agricultural Derivatives
- EEX Freight Derivatives
- EEX Indices data

The available data types are listed below – full details of the returned data for these symbols and the parameters can be found in section 5.5.:

- End of Day
- Intraday
- Trades
- Historic Data
- Best Bid and Ask
- Product Master Data

5. Market Data API

Criterion	Description
URL	All API request address: <ul style="list-style-type: none"> https://api1.datasources.eex-group.com
Data Domain	The API provides access to the Market Data (Time Series) stored in the EEX Database. This covers the following data areas: <ul style="list-style-type: none"> Power Natural Gas Freight Environmentals Agriculturals Indices
Availability	Market Data and the API are available for customers 24/7. The return formats provided are XML, CSV and JSON.
Source of Data	EEX Group trading, clearing and backend systems
Data Depth	Depending on the request parameters, the API will return latest data updates and historical data. The API will provide the latest EEX data available and is the quickest medium to receive prices, ideally moments after a price is known in the source system.
Data Coverage Period	Complete historical data is available starting from 01 January 2019. For some products, earlier data may be accessible, but completeness cannot be guaranteed for dates prior to 01 January 2019.

5.1 Request Method

The API is a synchronous interface. It supports one method of requesting data:

- Https Get**

Any date or filter parameters are a part of the URL query string that is submitted. In general, the request string is built as follows:

`https://api1.datasources.eex-group.com/<method>?` followed by “Parameter Name”=“Parameter Value”, where each pair of parameter name and value are separated by an ampersand (“&”)

Nine methods are supported:

- getAuction
- getDerivative
- getSpot
- getIndex
- getIntraday
- getHistory
- getPermission
- getProduct
- getQuote

5.2 Symbols

Symbology is the interpretation of how symbols are configured. Knowing the symbology, will assist you when creating a quote query or when configuring an historical table.

Formatting EEX Group Market Data

We use four digit shortcodes within all DataSource products to reference the different roots of the API. To find out the code for each contract, you can either use our getProduct call or have a look into our EEX Products (Shortcodes, WKN, ISIN, Vendor Codes, GoO) file.

EEX Group Market Data Symbols are composed as follows:

/E.PRODUCTCODE FOR EMISSION SPOT AND ALL FUTURES
 -E.PRODUCTCODE FOR OPTIONS
 #E.PRODUCTCODE FOR GAS SPOT
 #E.PRODUCTCODE FOR INDICES

If you want to address a specific maturity, you must address a product instead of a root. This is how the product codes are built:

Example:

/E.DEBMH20

- o DEBM = Phelix DE Base Load Month Futures
- o H = March
- o 20= 2020

EEX Month Codes

Month	Month Short Code
January	F
February	G
March	H
April	J
May	K
June	M
July	N
August	Q
September	U
October	V
November	X
December	Z

As the Get request is a URL string, special characters will need to be escaped. Escaping means to replace the special character with the escape symbol. For instance, to replace the space in 'a link', you must use this 'a%20link' in the URL. The most common of these special characters including their respective translation can be found in the table below.

The above-mentioned example would lead to this request (trade date is just one of the possible examples):

`https://api1.datasources.eex-group.com/getDerivatives/json?returnType=results&productType=futures&tradeDate=2020-02-06&product=%2FE.DEBMH20`

Character	Escape Symbol	Character	Escape Symbol
Space	%20	,	%2C
!	%21	/	%2F
#	%23	:	%3A
\$	%24	;	%3B
&	%26	=	%3D
'	%27	?	%3F
(%28	@	%40
)	%29	[%5B
*	%2A]	%5D
+	%2B		

This is not a comprehensive list. There are many resources on the internet that provide lookups for translation of URLs or specific special characters.

Special case for weekly contracts

For all weekly contracts (week, weekend), a zero must be added between the letters and the number that represents the week, if the data is requested on a product level.

Example:

- 1DB1 → EEX Danish DK1 Power Base Week Future
- 1st week in August 2024 → 1DB1Q24
- Add zero → 1DB01Q24

- <https://api1.datasource.eex-group.com/getDerivatives?&producttype=futures&tradedate=2024-07-10&returntype=results&product=%2FE.1DB01Q24>

This is not necessary when requesting the data on a root level.

Example:

<https://api1.datasource.eex-group.com/getDerivatives?&producttype=futures&tradedate=2024-07-10&returntype=results&Root=1DB1>

[https://api1.datasource.eex-](https://api1.datasource.eex-group.com/getDerivatives?&producttype=futures&tradedate=2024-07-10&returntype=results&Root=1DB1)

5.3 EndPoint

The endpoint for the service is:

- <https://api1.datasource.eex-group.com>

Please note, the endpoint supports html requests only. Furthermore, https (secure http) addresses are supported.

5.4 Authentication

Basic authentication is used for the API. This includes an EEX ID (example: EEX_1234) and a provided password (example: PassworD1234!). Please note that you cannot use your mail address or webshop login data for authentication, as this is not linked to the API. If you have problems while logging in, please contact us at datasource@eex-group.com or access.datasource@eex-group.com.

The User ID and password will be provided by EEX once a subscription is in place. If you already have an active subscription for other DataSource Software, the same password and ID are also used for the API.

5.5 Definition of API Formats

The following formats will be used for the data fields of the API.

Term	Description	Example
<datetime>	Time format – information of a point in time. Please note that all points in time are in UTC.	2017-01-17T01:00:00Z
<string>	Alphanumeric string – used for text information.	Nuclear
<integer>	The integer data type is used to specify a numeric value without a fractional component.	99
<float>	The float data type stores double-precision floating-point numbers with up to 17 significant digits.	34,15
<time>	Time format – information of a point in time. Please note that all points in time are in UTC.	15:05:00
<date>	Time format – information of a point in time. Please note that all points in time are in UTC.	2010-03-01
<real>	The real data type is used in to represent an approximation of a real number.	128,91
<period>	The period data type is an anchored duration. It represents a set of contiguous time granules within that duration and will consist of a beginning and ending bound.	2017-08

5.6 Definition of API Data Fields

This is a full list of the data fields and their types for reference.

Fieldname	Format	Description	Example
AuctionClearingPrice	<float>	The final clearing price for an emission primary auction, also known as “auction price”.	40,67
AuctionName	<string>	Name of the auction	EU
AuctionVolume	<integer>	Volume auctioned	735450
AverageBidSize	<float>	Average bid size	7354,50
AverageNumberOfBidsPerBidder	<integer>	Average number of Bids per bidder	2
AverageVolumeBidPerBidder	<integer>	Average volume bid per bidder	2222
AverageVolumeWonPerBidder	<integer>	Average volume won per bidder	3333
Contract	<string>	Contract of emission primary auction	T3PA
CountryRevenue	<integer>	Revenue per country of the auction	AT:318600
CoverRatio	<float>	Cover ratio of the auction	3,46
DeliveryEnd	<datetime>	Timestamp of delivery end date of a contract	2023-01-01T23:00:00Z
DeliveryPeriod	<period>	Delivery period of a derivatives contract. Please note that Options can have differing delivery periods. For example, a Year Option can have a Quarter as delivery period as it expires every 3 months.	Month, Quarter Year
DeliveryStart	<datetime>	Timestamp of delivery start date of a contract	2023-01-31T23:00:00Z
ExpiryDate	<datetime>	Timestamp of last settlement price amendment (if last day of delivery is on a weekend, the expiry date is on the next weekday)	2023-01-30T00:00:00Z
FrontContract	<period>	Front contract of the index (yyyy-mm)	2017-08
HighPrice	<real>	Highest prices of the trading day	128,91
LastPrice	<real>	Price of the last trade on the trading day	45,95
LongName	<string>	Plain text of the product code	Phelix-DE Base Month Future
LotSize	<integer>	Volume traded for the contract in the product’s commodity unit	18100
LowPrice	<real>	Lowest price of the trading day	12,07
MarketArea	<string>	Name of the natgas market area (Detailed information may be found in the contract specification).	TTF
Maturity	<period>	Maturity of the products displays the month in which the delivery starts for long term contracts (month, quarter, season, year) displays the month in with the	2017-04

Fieldname	Format	Description	Example
		delivery ends for short term contracts (week, weekend).	
MaturityType	<string>	A filter and return item which denotes the periodicity of a Product. DAY = Day Contracts WEEKEND = Weekend Contracts WEEK = Week contracts MONTH = month contracts QUARTER = quarter contracts SEASON = season contracts YEAR = year contracts	DAY
MaximumBid	<float>	Maximum bid of the auction	6,00
Mean	<float>	Mean price of the auction	5,54
Median	<float>	Median price of the auction	5,54
MinimumBid	<float>	Minimum bid of the auction	5,54
NumberOfBidsSubmitted	<integer>	Number of submitted bids	51
NumberOfSuccessfulBids	<integer>	Number of successful bids	4
NumberOfTrades	<integer>	Total number of trades per trading day	25
OpenInterestLots	<integer>	Open interest lots	12285
OpenInterestVolume	<integer>	Open interest lots multiplied by lot size.	234568
OpenPrice	<float>	Price of the first trade at an exchange market.	42,67
Product	<string>	Code of the product	/E.DB18N19
Root	<string>	Code of a chain of EEX short codes	H9BQ
SettlementPrice	<float>	A settlement price is determined for each individual contract which can be traded continuously or Trade Registration on EEX Power Derivatives and on the EEX Derivatives and Spot Markets every day.	42,67
StandardDeviationofBidVolumePerBidder	<integer>	Standard Deviation derived from the Bid Volume Per Bidder	15
Status	<string>	Statement whether an auction has been successfully conducted.	Successful
Strike	<integer>	Strike price of the option	9000
Timestamp (WDRP)	<datetime>	Shows the day of the delivery.	2024-07-10T00:00:00Z
TimestampHighPrice	<datetime>	Timestamp of trade with the highest price of the day	2017-06-14T13:45:00Z
TimestampLastPrice	<datetime>	Timestamp of last trade of trading day	2017-06-14T16:45:00Z
TimestampLowPrice	<datetime>	Timestamp of trade with the lowest price of the day	2017-06-14T11:45:00Z
TimestampOpenPrice	<datetime>	Timestamp of first trade at an exchange market or the first registered traded price	2017-06-14T10:45:00Z

Fieldname	Format	Description	Example
		within business hours of EEC for trade reg. (OTC)	
TotalNumberOfBidders	<integer>	Total number of bidders of the auction	16
TotalRevenue	<float>	Total revenue of the auction	11050700
TotalVolumeOfBidsSubmitted	<integer>	Number of successful bids of the auction	1211
TradeDate	<date>	Date of Trading	2014-09-20
TradedLots	<integer>	Number of traded contracts/lots	58
TradedVolume	<integer>	Traded Contracts multiplied by contract volume	4200
TradeID	<string>	Identifier of a trade	30840
TradeTimestamp	<datetime>	Date and time of the trade	2017-06-14T16:45:00Z
TradingEnd	<datetime>	Date and time of end of trading period	2023-01-29T23:00:00Z
TradingStart	<datetime>	Date and time of start of trading period	2022-03-30T22:00:00Z
Type	<string>	Field shows if a trade was a trade registration (OTC) or an exchange trade	EXCHANGE TRADEREG
Underlying	<string>	Underlying contract of the product (Detailed information may be found in the contract specification)	F1BQ
UnitOfPrices	<string>	Contains information on currency (Detailed information may be found in the contract specification)	EUR
UnitofVolumes	<string>	Contains information on measuring unit (Detailed information may be found in the contract specification)	MWh
Valid	<string>	Traded version of the trade (If valid then "yes" if invalid then "no"). A trade is invalid if it has been cancelled	Yes or No
ValidTrade	<string>	Traded version of the trade (If valid then "yes" if invalid then "no"). A trade is invalid if it has been cancelled	Yes or No
Volume	<integer>	Volume traded for the contract	537189

5.8 Market Data API functions, available fields, usage and return format

The following chapter lists all available data types and the fields that the API can return. The heading for each section is the function call as passed into the API.

5.8.1 Principles

The following principles shall be observed when developing or maintaining calls for automated use of data. This applies to all existing and new users in the same way.

- The number of concurrent sessions by a single UserID should not exceed 10.
- The total number of requests by a single UserID should not exceed 25,000 per day or rolling 24-hour period.
- Single UserID payload size limits are as follows:
 - Less than 100 calls to the same method within a single hour.
 - No more than 3 products within a single call.
 - No more than 18 requests per minute.
- Single UserID payload size limit for getHistory are as follows:
 - Less than 2 years of history within a single call for daily granularity per product.
 - Less than 1 week of history within a single call for hourly granularity per product.

In the event of observed non-compliance with the above restrictions, EEX reserves the right to block UserID in the interest of all other users according to 4.1.1.4 of the General Terms and Conditions.

5.8.2 getPermissions

getPermissions returns all symbols permissioned for the user. This includes current and historical products. Further information about the products including meta data is available via getProduct or can be found in our EEX Products (Shortcodes, WKN, ISIN, Vendor Codes, GoO) file.

Parameters

Name	Optional	Description
Fields	Yes	A list of fields to return - available fields are listed below. If no field is selected, all fields get returned
Return Format	Yes	If no return format is added, the default return format is XML JSON - return format is json XML – return format is xml CSV – return format is csv
Commodity	Yes	Identifier or a comma separated list of identifiers for the commodity required Valid Arguments are: POWER – Power Derivatives NATGAS – Natural Gas Spot and Derivatives AGRI – Agricultural Futures EMISSIONS – Emissions Auction, Spot and Derivatives FREIGHT – Freight Futures

Name	Optional	Description
		HYDROGEN – Hydrix GO – GO Futures

Available Fields

- Product
- Maturity
- LongName
- Root
- Underlying
- FrontContract
- Commodity
- MarketArea

Usage and Return Sample

<https://api1.datasource.eex-group.com/getPermissions/json?commodity=EMISSIONS>

```
{ "results": [
{ "result": [
{
"Root": "FEAA",
"Underlying": "",
"Commodity": "EMISSIONS",
"MarketArea": ""
},
{...

```

If you are not interested in a specific commodity like Power or Natgas, but want to see all permissioned roots or specific indices like the EGSI or HYDRIX, please use this request:

<https://api1.datasource.eex-group.com/getPermissions/json>

5.8.3 getProduct

getProduct returns the metadata associated with a specific symbol or list of symbols.

Parameters

Name	Optional	Description
Root	No	A precise identifier or a comma separated list of identifiers for the list of required products must be provided
Fields	Yes	A list of fields to return - available fields are listed below. If no field is selected, all fields get returned
Return Format	Yes	If no return format is added, the default return format is XML JSON - return format is json XML – return format is xml CSV – return format is csv

Available Fields

- Product
- Root
- LongName
- MarketArea
- Commodity
- DeliveryPeriod
- DeliveryStart
- DeliveryEnd
- Maturity
- Underlying
- Type
- FrontContract
- Strike
- ExpiryDate
- UnitofPrices
- UnitofVolumes
- TradingStart
- TradingEnd

Usage and Return Sample

<https://api1.datasource.eex-group.com/getProduct/xml?root=DEBM>

```
<Row>
<Field Name="Product">/E.DEBMF24</Field>
<Field Name="Root">DEBM</Field>
<Field Name="LongName">EEX German Power Base Month Future</Field>
<Field Name="MarketArea"/>
<Field Name="Commodity">POWER</Field>
<Field Name="DeliveryPeriod">MONTH</Field>
<Field Name="DeliveryStart">2024-01-01T23:00:00Z</Field>
<Field Name="DeliveryEnd">2024-01-31T23:00:00Z</Field>
<Field Name="Maturity">2024-01</Field>
<Field Name="Underlying"/>
<Field Name="Type">FUTURE</Field>
<Field Name="FrontContract">2023-04</Field>
<Field Name="Strike"/>
<Field Name="ExpiryDate">2024-01-30T00:00:00Z</Field>
<Field Name="UnitofPrices">EUR</Field>
<Field Name="UnitofVolumes">MWh</Field>
<Field Name="TradingStart">2023-03-30T22:00:00Z</Field>
<Field Name="TradingEnd">2024-01-29T23:00:00Z</Field>
</Row>
...
```

5.8.4 getDerivatives

This function returns settlements or trade results data for a specific day for a specific commodity, root or product. It can be filtered by Market Area/Country where applicable.

Parameters

Name	Optional	Description
Commodity	Yes	Identifier or a comma separated list of identifiers for the symbols: Valid Arguments are: POWER – Power Futures and Options NATGAS – Natural Gas Futures and Options AGRI – Agricultural Futures and Options EMISSIONS – Emissions Futures and Options FREIGHT – Freight Futures GO – GO Futures To select products to return – either a commodity, list of products or root(s) must be defined.
Root	Yes	Precise identifier or a comma separated list of identifiers for the products required
Products	Yes	Identifier or a comma separated list of identifiers for the products required

Name	Optional	Description
ReturnType	No	Filters are RESULTS or TRADES. Results are the pricing fields for the selected products, trades are the list of trades for the selected products. Either RESULTS or TRADES must be provided.
ProductType	No	Either valid filters FUTURES or OPTIONS must be provided.
MarketArea	Yes	Market Area – example TTF for Natural Gas. If no market area is provided or market area is not applicable all instruments for the commodity group will be returned
TradeDate	No	A date in the format 'YYYY-MM-DD' representing the trading date must be provided
Fields	Yes	A list of fields to return - available fields are listed below. If no field is selected, all fields get returned
Return Format	Yes	If no return format is added, the default return format is XML JSON - return format is json XML – return format is xml CSV – return format is csv
MaturityType	Yes	Filters for maturity. Valid inputs are: D = Day Contracts W = Week contracts M = month contracts Y = year contracts

Available Fields and Usage

ReturnType=Results

- Product
- LongName
- Root
- Maturity
- MaturityType
- MarketArea
- Strike
- DeliveryStart
- DeliveryEnd
- OpenPrice
- TimestampOpenPrice
- HighPrice
- TimestampHighPrice
- LowPrice
- TimeStampLowPrice

- LastPrice
- TimeStampLastPrice
- SettlementPrice
- UnitofPrices
- LotSize_Exch
- TradedLots_Exch
- NumberofTrades_Exch
- TradedVolume_Exch
- LotSize_TR
- TradedLots_TR
- NumberofTrades_TR
- TradedVolume_TR
- OpenInterestLots
- OpenInterestVolume
- UnitofVolume

Note: The Return Type=Results always returns the tradable products that received a settlement price on the requested trading day.

Usage and Return Sample

<https://api1.datasource.eex-group.com/getDerivatives/json?returnType=results&productType=futures&tradeDate=2023-04-04&root=ATBY>

```
{ "results": [
  { "result": [
    {
      "Product": "\/E.ATBYF24",
      "LongName": "EEX Austrian Power Base Year Future",
      "Root": "ATBY",
      "Maturity": "2024-01",
      "MaturityType": "YEAR",
      "MarketArea": "",
      "Strike": "",
      "DeliveryStart": "2024-01-01T23:00:00Z",
      "DeliveryEnd": "2024-12-31T23:00:00Z",
      "OpenPrice": "168,25",
      "TimestampOpenPrice": "2023-04-04T12:24:34Z",
      "HighPrice": "168,25",
      "TimestampHighPrice": "2023-04-04T12:24:34Z",
      "LowPrice": "168,25",
      "TimestampLowPrice": "2023-04-04T12:24:34Z",
      "LastPrice": "168,25",
      "TimestampLastPrice": "2023-04-04T12:24:34Z",
```

```
"SettlementPrice": "163,90",
"UnitofPrices": "EUR",
"LotSize_Exch": 8784,
"TradedLots_Exch": 1,
"NumberOfTrades_Exch": 1,
"TradedVolume_Exch": 8784,
"LotSize_TR": 8784,
"TradedLots_TR": 3,
"NumberOfTrades_TR": 3,
"TradedVolume_TR": 26352,
"OpenInterestLots": 109,
"OpenInterestVolume": "957456",
"UnitofVolumes": "MWh"
},...
```

Other examples:

<https://api1.datasource.eex-group.com/getDerivatives/xml?returnType=results&productType=futures&tradeDate=2022-05-03&commodity=natgas>

<https://api1.datasource.eex-group.com/getDerivatives/xml?returnType=results&productType=futures&tradeDate=2022-05-03&Product=%2FE.FEUAZ22>

<https://api1.datasource.eex-group.com/getDerivatives/json?returnType=results&productType=options&tradeDate=2022-05-06&root=O2BY>

Return Type=Trades

- Product
- Long Name
- Root
- Maturity
- MaturityType
- MarketArea
- Type
- Underlying
- Strike
- DeliveryStart
- DeliveryEnd
- TradedTimeStamp
- TradeID
- Valid
- Price
- UnitofPrices
- LotSize
- TradedLots
- TradedVolume

- UnitofVolumes
- TradedType

Usage and Return Sample

<https://api1.datasource.eex-group.com/getDerivatives/json?returnType=trades&productType=futures&tradeDate=2023-03-16&root=DEBY>

```
{ "results": [
  { "result": [
    {
      "Product": "\/E.DEBYF24",
      "LongName": "EEX German Power Base Year Future",
      "Root": "DEBY",
      "Maturity": "2024-01",
      "MaturityType": "YEAR",
      "MarketArea": "",
      "Type": "EXCHANGE",
      "Underlying": "",
      "Strike": "",
      "DeliveryStart": "2024-01-01T23:00:00Z",
      "DeliveryEnd": "2024-12-31T23:00:00Z",
      "TradeTimeStamp": "2023-03-16T16:16:37Z",
      "TradeID": "14801920",
      "Valid": "Yes",
      "Price": "136,00",
      "UnitofPrices": "EUR",
      "LotSize": 8784,
      "TradedLots": 4,
      "TradedVolume": 35136,
      "UnitofVolumes": "MWh",
      "TradedType": "FUTURES"
    },
    { ...
  ]
}
```

5.8.5 getSpot

This function returns spot results data of a specific day for gas and emissions. Results or trades can be retrieved for specific commodities, roots or products.

Parameters

Name	Optional	Description
Commodity	No	Identifier or a comma separated list of identifiers for the symbols required Valid Arguments are NATGAS – Natural Gas EMISSIONS – Emissions Either NATGAS or EMISSIONS must be provided.
Root	Yes	Precise identifier or a comma separated list of identifiers for the products required

Name	Optional	Description
Products	Yes	Precise identifier or a comma separated list of identifiers for the products required
ReturnType	No	Either filter RESULTS or TRADES must be provided. Results are the pricing fields for the selected Products, Trades are the list of trades for the selected products.
MarketArea	Yes	Only applicable for natgas contracts. If no market area is supplied or market area is not applicable all instruments for the commodity group will be returned
TradeDate	No	A date in the format 'YYYY-MM-DD' must be provided.
Fields	Yes	A list of fields to return - available fields are listed below
Return Format	Yes	If no return format is added, the default return format is XML JSON - return format is json XML – return format is xml CSV – return format is csv

Available Fields

Gas Results

- MarketArea
- Root
- LongName
- DeliveryStart
- DeliveryEnd
- OpenPrice
- TimestampOpenPrice
- HighPrice
- TimestampHighPrice
- LowPrice
- TimestampLowPrice
- LastPrice
- TimestampLastPrice
- SettlementPrice
- UnitofPrices
- TradedLots
- NumberofTrades
- TradedVolume
- UnitOfVolumes

Usage and Return Sample

<https://api1.datasources.eex-group.com/getSpot/xml?returnType=results&commodity=NATGAS&marketarea=tff&&tradeDate=2022-12-09>

```

...
<Row>
<Field Name="MarketArea">TTF</Field>
<Field Name="Root">GND1</Field>
<Field Name="LongName">DAY 1 MW</Field>
<Field Name="DeliveryStart">2022-12-12T05:00:00Z</Field>
<Field Name="DeliveryEnd">2022-12-13T04:59:59Z</Field>
<Field Name="OpenPrice">136,000</Field>
<Field Name="TimestampOpenPrice">2022-12-09T07:43:17Z</Field>
<Field Name="HighPrice">148,500</Field>
<Field Name="TimestampHighPrice">2022-12-09T16:56:25Z</Field>
<Field Name="LowPrice">136,000</Field>
<Field Name="TimestampLowPrice">2022-12-09T07:43:17Z</Field>
<Field Name="LastPrice">146,500</Field>
<Field Name="TimestampLastPrice">2022-12-09T22:11:36Z</Field>
<Field Name="SettlementPrice">143,215</Field>
<Field Name="UnitOfPrices">EUR</Field>
<Field Name="TradedLots">161258</Field>
<Field Name="NumberOfTrades">3107</Field>
<Field Name="TradedVolume">3870192</Field>
<Field Name="UnitOfVolumes">MWh</Field>
</Row>
...

```

Gas Trades

Available Fields

- Root
- LongName
- MarketArea
- TradeTimeStamp
- DeliveryStart
- DeliveryEnd
- TradeID
- ValidTrade
- Price
- UnitofPrices
- LotSize
- TradedLots
- TradedVolume
- UnitofVolumes

- TradedType

Usage and Return Sample

[https://api1.datasources.eex-](https://api1.datasources.eex-group.com/getSpot/xml?returnType=trades&commodity=NATGAS&marketarea=the&root=GND1&tradeDate=2023-04-04)

[group.com/getSpot/xml?returnType=trades&commodity=NATGAS&marketarea=the&root=GND1&tradeDate=2023-04-04](https://api1.datasources.eex-group.com/getSpot/xml?returnType=trades&commodity=NATGAS&marketarea=the&root=GND1&tradeDate=2023-04-04)

```
<Results>
<Result>
<Row>
<Field Name="Root">GND1</Field>
<Field Name="LongName">DAY 1 MW</Field>
<Field Name="MarketArea">THE</Field>
<Field Name="TradeTimestamp">2023-04-04T00:10:30Z</Field>
<Field Name="DeliveryStart">2023-04-04T04:00:00Z</Field>
<Field Name="DeliveryEnd">2023-04-05T04:00:00Z</Field>
<Field Name="TradeID">1159156300</Field>
<Field Name="ValidTrade">Yes</Field>
<Field Name="Price">53,000</Field>
<Field Name="UnitOfPrices">EUR</Field>
<Field Name="LotSize">24</Field>
<Field Name="TradedLots">100</Field>
<Field Name="TradedVolume">2400</Field>
<Field Name="UnitOfVolumes">MWh</Field>
<Field Name="TradedType">EXCHANGE</Field>
</Row>
...
```

Emission Trades

Available Fields

- Product
- Root
- LongName
- MarketArea
- TradeTimestamp
- DeliveryPeriod
- TradeID
- ValidTrade
- Price
- UnitofPrices
- TradedVolume
- UnitofVolumes
- TradedType

Usage and Return Sample

[https://api1.datasource.eex-](https://api1.datasource.eex-group.com/getSpot/xml?returnType=trades&commodity=emissions&root=SEME&tradeDate=2023-03-16)

[group.com/getSpot/xml?returnType=trades&commodity=emissions&root=SEME&tradeDate=2023-03-16](https://api1.datasource.eex-group.com/getSpot/xml?returnType=trades&commodity=emissions&root=SEME&tradeDate=2023-03-16)

```
<Results>
<Result>
<Row>
<Field Name="Product">/E.SEMEZ29</Field>
<Field Name="Root">SEME</Field>
<Field Name="LongName">EEX EUA Spot</Field>
<Field Name="MarketArea">EU</Field>
<Field Name="TradeTimestamp">2023-03-16T08:11:05Z</Field>
<Field Name="DeliveryPeriod"/>
<Field Name="TradeID">691200</Field>
<Field Name="ValidTrade"/>
<Field Name="Price">88,000</Field>
<Field Name="UnitOfPrices">EUR</Field>
<Field Name="TradedVolume">1000</Field>
<Field Name="UnitOfVolumes">tCO2</Field>
<Field Name="TradedType">EXCHANGE</Field>
</Row>
...
```

5.8.6 getAuction

This function returns emission auction results data for a specific day.

Parameters

Name	Optional	Description
TradeDate	No	A date in the format 'YYYY-MM-DD' must be provided
Return Format	Yes	If no return format is added, the default return format is XML JSON - return format is json XML – return format is xml CSV – return format is csv

Available Fields

- Time
- AuctionName
- Contract
- Status
- AuctionClearingPrice
- MinimumBid

- MaximumBid
- Mean
- Median
- UnitOfPrices
- AuctionVolume
- TotalVolumeOfBidsSubmitted
- NumberOfBidsSubmitted
- NumberOfSuccessfulBids
- AverageNumberOfBidsPerBidder
- AverageBidSize
- AverageVolumeBidPerBidder
- StandardDeviationOfBidVolumePerBidder
- AverageVolumeWonPerBidder
- StandardDeviationOfVolumeWonPerBidder
- CoverRatio
- TotalNumberOfBidders
- NumberOfSuccessfulBidders
- TotalRevenue
- CountryRevenue

Usage and Return Sample

<https://api1.datasource.eex-group.com/getAuction/xml?tradeDate=2023-02-13>

```
<Results>
<Result>
<Row>
<Field Name="Time">10:00:01</Field>
<Field Name="AuctionName">EEX EUA Primary Auction Spot</Field>
<Field Name="Contract">T3PA</Field>
<Field Name="Status">Successful</Field>
<Field Name="AuctionClearingPrice">88,86</Field>
<Field Name="MinimumBid">70</Field>
<Field Name="MaximumBid">100</Field>
<Field Name="Mean">88,9</Field>
<Field Name="Median">89,09</Field>
<Field Name="UnitOfPrices">EUR</Field>
<Field Name="AuctionVolume">2409000</Field>
<Field Name="TotalVolumeOfBidsSubmitted">4145000</Field>
<Field Name="NumberOfBidsSubmitted">70</Field>
<Field Name="NumberOfSuccessfulBids">42</Field>
<Field Name="AverageNumberOfBidsPerBidder">4,12</Field>
```

```

<Field Name="AverageBidSize">59214,286</Field>
<Field Name="AverageVolumeBidPerBidder">243823</Field>
<Field Name="StandardDeviationOfBidVolumePerBidder">389042</Field>
<Field Name="AverageVolumeWonPerBidder">150562</Field>
<Field Name="StandardDeviationOfVolumeWonPerBidder">244475</Field>
<Field Name="CoverRatio">1,72</Field>
<Field Name="TotalNumberOfBidders">17</Field>
<Field Name="NumberOfSuccessfulBidders">16</Field>
<Field Name="TotalRevenue">214063740</Field>
<Field Name="CountryRevenue">EU</Field>
</Row>

```

...

5.8.7 getIndex

This function returns index results for a specific day

Parameters

Name	Optional	Description
IndexType	No	The Index for which results should be returned. Valid tokens are <ul style="list-style-type: none"> • AG (Agriculture Indices) • APDD • ECARBIX • EGIX • EGSi • HYDRiX • MONTHLY • NGP (currently not available) • WDRP
MarketArea	Yes	Where applicable (for instance EGIX) a market area(s) can be passed to limit the returned products
TradeDate	No	A date in the format 'YYYY-MM-DD' must be provided.
DeliveryDate	No	A date in the format 'YYYY-MM-DD' must be provided.

Name	Optional	Description
		Please note: DeliveryDate is only used as parameter for APDD index.
Fields	Yes	A list of fields to return - available fields are listed below
Return Format	Yes	If no return format is added, the default return format is XML JSON - return format is json XML – return format is xml CSV – return format is csv

Available Fields

Agricultural Index

- Longname
- Product
- ReferenceWeek
- Market Area
- Price
- UnitofPrices

Usage and Return Sample

<https://api1.datasource.eex-group.com/getIndex/xml?indexType=AG&tradeDate=2023-02-16>

```
<Row>
<Field Name="LongName">Weekly Dutch European Processing Potato Index</Field>
<Field Name="Product">$E.FAPP_NL</Field>
<Field Name="ReferenceWeek">Week 07/23</Field>
<Field Name="Price">30,130</Field>
<Field Name="UnitofPrices">EUR/100kg</Field>
</Row>
```

APDD

- MarketArea
- Timestamp
- MarginalBuyPrice
- MarginalSellPrice
- WeightedAveragePrice
- UnitOfPrice
- Volume
- DeliveryStart
- DeliveryEnd
- UnitofVolume

Usage and Return Sample

<https://api1.datasource.eex-group.com/getIndex/?IndexType=APDD&DeliveryDate=2022-11-06&MarketArea=THE>

```
<Results>
<Result>
<Row>
<Field Name="MarketArea">THE</Field>
<Field Name="Timestamp">2022-11-07T02:00:07Z</Field>
<Field Name="MarginalBuyPrice"/>
<Field Name="MarginalSellPrice">56,000</Field>
<Field Name="WeightedAveragePrice">58,404</Field>
<Field Name="UnitofPrice">EUR</Field>
<Field Name="Volume">4531514</Field>
<Field Name="DeliveryStart">2022-11-06T05:00:00Z</Field>
<Field Name="DeliveryEnd">2022-11-07T05:00:00Z</Field>
<Field Name="UnitofVolume">MWh</Field>
</Row>
</Result>
</Results>
```

ECARBIX

- Longname
- Product
- IndexType (as in Day, Month, Year)
- Price
- Volumes
- UnitofPrices
- UnitofVolume
- TradeDate

After 1.1.2024, the field “Volumes” will be empty. This is an expected behavior.

Usage and Return Sample

<https://api1.datasource.eex-group.com/getIndex/xml?indexType=ECARBIX&tradeDate=2023-02-28>

```
<Results>
<Result>
<Row>
<Field Name="LongName">EEX ECarbix Month Index</Field>
<Field Name="Product">#E.ECARBIX.M</Field>
<Field Name="IndexType">Month</Field>
<Field Name="Price">91,933</Field>
<Field Name="Volumes">39454000</Field>
<Field Name="UnitofPrices">EUR</Field>
<Field Name="UnitofVolume">tCO2</Field>
<Field Name="TradeDate">2023-02-28</Field>
</Row>
<Row>
<Field Name="LongName">EEX ECarbix Day Index</Field>
<Field Name="Product">#E.ECARBIX.D</Field>
<Field Name="IndexType">Day</Field>
<Field Name="Price">96,330</Field>
<Field Name="Volumes">2427000</Field>
<Field Name="UnitofPrices">EUR</Field>
<Field Name="UnitofVolume">tCO2</Field>
<Field Name="TradeDate">2023-02-28</Field>
</Row>
</Result>
</Results>
```

EGIX

- LongName
- IndexType (as in Day, Month)
- MarketArea
- FrontContract
- Price
- Volumes
- UnitofPrices
- UnitofVolume
- TradeDate
-

Usage and Return Sample

[https://api1.datasources.eex-](https://api1.datasources.eex-group.com/getIndex/xml?indexType=EGIX&marketarea=TTF&tradeDate=2023-03-08)

[group.com/getIndex/xml?indexType=EGIX&marketarea=TTF&tradeDate=2023-03-08](https://api1.datasources.eex-group.com/getIndex/xml?indexType=EGIX&marketarea=TTF&tradeDate=2023-03-08)

```

<Results>
<Result>
<Row>
<Field Name="LongName">EGIX-Day</Field>
<Field Name="IndexType">Day</Field>
<Field Name="MarketArea">TTF</Field>
<Field Name="FrontContract">2023-04</Field>
<Field Name="Price">42,868</Field>
<Field Name="Volumes">0</Field>
<Field Name="UnitofPrices">EUR</Field>
<Field Name="UnitofVolume">MWh</Field>
<Field Name="TradeDate">2023-03-08</Field>
</Row>
<Row>
<Field Name="LongName">EGIX-Month</Field>
<Field Name="IndexType">Month</Field>
<Field Name="MarketArea">TTF</Field>
<Field Name="FrontContract">2023-04</Field>
<Field Name="Price">45,136</Field>
<Field Name="Volumes">0</Field>
<Field Name="UnitofPrices">EUR</Field>
<Field Name="UnitofVolume">MWh</Field>
<Field Name="TradeDate">2023-03-08</Field>
</Row>
</Result>
</Results>

```

EGSI

- IndexType
- MarketArea
- Longname
- Timestamp
- DeliveryStart
- DeliveryEnd
- Price
- UnitofPrices

When requesting the EGSI, you must address the (first) day of delivery in the query. For EGSI Day it's the day of the respective delivery, for EGSI Weekend, Week and Month the first day of the respective delivery, e.g. the date of Monday for EGSI Week, Saturday for EGSI Weekend and the first calendar day of the month for EGSI Month.

Usage and Return Sample

<https://api1.datasources.eex-group.com/getIndex/xml?indexType=EGSI&DeliveryDate=2023-10-11&MarketArea=TTF>

```
<Results>
<Result>
<Row>
<Field Name="IndexType">Day</Field>
<Field Name="MarketArea">TTF</Field>
<Field Name="LongName">EEX EGSI TTF Day</Field>
<Field Name="Timestamp">2023-10-10T00:00:00Z</Field>
<Field Name="DeliveryStart">2023-10-11T04:00:00Z</Field>
<Field Name="DeliveryEnd">2023-10-12T04:00:00Z</Field>
<Field Name="Price">44,751</Field>
<Field Name="UnitofPrices">EUR/MWh</Field>
</Row>
</Result>
</Results>
```

HYDRIX

- LongName
- Product
- MarketArea
- IndexType
- Price
- UnitofPrices
- TradeDate

Usage and Return Sample

<https://api1.datasources.eex-group.com/getIndex/xml?indexType=hydrix&tradeDate=2023-08-16>

```

<Results>
<Result>
<Row>
<Field Name="LongName">EEX Green Weekly HYDRIX DE</Field>
<Field Name="MarketArea">DE</Field>
<Field Name="Product">$E.GNHYDRIX.W.DE</Field>
<Field Name="IndexType">Week</Field>
<Field Name="Price">38,228</Field>
<Field Name="UnitofPrices">EUR/MWh</Field>
<Field Name="TradeDate">2023-08-16</Field>
</Row>
<Row>
<Field Name="LongName">EEX Green Weekly HYDRIX NL</Field>
<Field Name="MarketArea">NL</Field>
<Field Name="Product">$E.GNHYDRIX.W.NL</Field>
<Field Name="IndexType">Week</Field>
<Field Name="Price">4,709</Field>
<Field Name="UnitofPrices">EUR/MWh</Field>
<Field Name="TradeDate">2023-08-16</Field>
</Row>
<Row>
<Field Name="LongName">EEX Grey Weekly HYDRIX DE</Field>
<Field Name="MarketArea">DE</Field>
<Field Name="Product">$E.GRHYDRIX.W.DE</Field>
<Field Name="IndexType">Week</Field>
<Field Name="Price">4,285</Field>
<Field Name="UnitofPrices">EUR/MWh</Field>
<Field Name="TradeDate">2023-08-16</Field>
</Row>
<Row>
<Field Name="LongName">EEX Grey Weekly HYDRIX NL</Field>
<Field Name="Market Area">NL</Field>
<Field Name="Product">$E.GRHYDRIX.W.NL</Field>
<Field Name="IndexType">Week</Field>
<Field Name="Price">4,586</Field>
<Field Name="UnitofPrices">EUR/MWh</Field>
<Field Name="TradeDate">2023-08-16</Field>
</Row>
</Result>
</Results>

```

MONTHLY

For the monthly index, you can choose any date of the respective month as the trade date.

- LongName
- Product
- MarketArea
- Price
- UnitofPrices
- UnitofVolume
- TradeDate

Usage and Return Sample

<https://api1.datasource.eex-group.com/getIndex/xml?indexType=monthly&tradeDate=2023-02-08>

```

<Row>
<Field Name="LongName">EEX CEGH Natural Gas Month Index</Field>
<Field Name="Product">$E.G8BM</Field>
<Field Name="MarketArea">CEGH</Field>
<Field Name="Price">54,778</Field>
<Field Name="UnitofPrices">EUR</Field>
<Field Name="UnitofVolume">MWh</Field>
<Field Name="TradeDate">2023-02-27</Field>
</Row>...

```

WDRP

- Longname
- Product
- Marketarea
- Price
- UnitofPrice
- UnitofVolume
- Timestamp
- DeliveryStart
- DeliveryEnd

Usage and Return Sample

<https://api1.datasource.eex-group.com/getIndex/xml?indexType=WDRP&deliveryDate=2023-08-04>

Similar to the EGSI, you can integrate an optional market area filter:

<https://api1.datasource.eex-group.com/getIndex/xml?indexType=WDRP&deliveryDate=2023-08-04&marketarea=TTF>

```
<Results>
<Result>
<Row>
<Field Name="LongName">EEX Within-Day Reference Price TTF</Field>
<Field Name="Product">$E.WDRP_TTF</Field>
<Field Name="MarketArea">TTF</Field>
<Field Name="Price">29,250</Field>
<Field Name="UnitofPrices">EUR</Field>
<Field Name="UnitofVolume">MWh</Field>
<Field Name="Timestamp">2023-08-06T00:00:00Z</Field>
<Field Name="DeliveryStart">2023-08-04T04:00:00Z</Field>
<Field Name="DeliveryEnd">2023-08-05T04:00:00Z</Field>
</Row>
</Result>
</Results>
```

5.8.8 getHistory

getHistory returns trade information across a range of days for a list of items. Please note that this call only includes historic trades. Trades from the current day are not included.

Parameters

Name	Optional	Description
Root	No	Identifier or a comma separated list of identifiers for the products must be provided.
Return Format	Yes	If no return format is added, the default return format is XML JSON - return format is json XML – return format is xml CSV – return format is csv
ProductType	No	Depending in which root or product gets addressed, the right product type must be chosen. Valid Arguments are: FUTURES – Futures of all commodities OPTIONS – Options of all commodities INDEX – all Indices SPOT – Gas and Emission spot contracts
Product	Yes	Precise identifier or a comma separated list of identifiers for the products required
Start	Yes	A date in the format, depending on the time resolution of the time series requested (e.g. 'YYYY-MM-DD hh:mm'). Is equal to the first output TimeStamp.

Name	Optional	Description
		If a start date is specified an end date must be specified. Otherwise the DaysBack parameter can be set. One or the other is required. If both are supplied the Start and End date will take precedence.
End	Yes	A date in the format, depending on the time resolution of the time series requested (e.g. 'YYYY-MM-DD hh:mm'). Is equal to the last output TimeStamp.
DaysBack	Yes	Synonymous with the START/ END pair, with the proviso that END is the current date. So 3, would count 3 days back from today.
Fields	Yes	Fields are dependent on the type of instrument to be retrieved, see the fields available in functions detailed in 5.6.1.2 to 5.6.1.14

Available Fields

The available fields are determined by the Product codes requested. The fields available in the functions GetDerivatives, GetSpot, GetIndex for Trades will be available.

Usage and Return Sample

<https://api1.datasources.eex-group.com/getHistory/xml?root=G0BM&ProductType=FUTURES&daysBack=1>

```
<Results>
<Result>
<Row>
<Field Name="Product">/E.G0BMK23</Field>
<Field Name="Root">G0BM</Field>
<Field Name="LongName">EEX THE Natural Gas Month Future</Field>
<Field Name="Maturity">2023-05</Field>
<Field Name="MaturityType">MONTH</Field>
<Field Name="MarketArea">THE</Field>
<Field Name="Type">EXCHANGE</Field>
<Field Name="Underlying"/>
<Field Name="Strike"/>
<Field Name="DeliveryStart">2023-05-01T22:00:00Z</Field>
<Field Name="DeliveryEnd">2023-05-31T22:00:00Z</Field>
<Field Name="TradeTimeStamp">2023-04-04T09:04:57Z</Field>
<Field Name="TradeID">5652480</Field>
<Field Name="Valid">Yes</Field>
<Field Name="Price">49,50</Field>
<Field Name="UnitofPrices">EUR</Field>
<Field Name="LotSize">744</Field>
<Field Name="TradedLots">10</Field>
<Field Name="TradedVolume">7440</Field>
<Field Name="UnitofVolumes">MWh</Field>
<Field Name="TradedType">FUTURE</Field>
</Row>
...
```

Another example:

<https://api1.datasource.eex-group.com/getHistory/json?product=/E.DEBYF23&ProductType=FUTURES&start=2022-01-31&end=2022-02-01>

<https://api1.datasource.eex-group.com/getHistory/xml?root=GNHYDRIX.W&ProductType=Index&daysBack=10>

5.8.9 getQuote

getQuote returns the current latest price information for specific code or list of codes, it returns only the latest price information. The most recently stored value is returned for each field requested.

Parameters

Name	Optional	Description
Product	No	Identifier or a comma separated list of identifiers for the products must be provided. Product or root is mandatory.
Return Format	Yes	If no return format is added, the default return format is XML JSON - return format is json XML – return format is xml CSV – return format is csv
Root	Yes	Precise identifier or a comma separated list of identifiers for the products required. Product or root is mandatory.
Fields	Yes	Fields are dependent on the type of instrument to be retrieved, see the fields available in functions detailed in 5.6.1.2 to 5.6.1.14

Available Fields

- Product
- Root
- BestBidPrice
- BestBidVolume
- TimestampBestBidPrice
- BestAskPrice
- BestAskVolume
- TimestampBestAskPrice
- OpenPrice
- TimestampOpenPrice
- HighPrice
- TimestampHighPrice
- LowPrice
- TimestampLowPrice
- LastPrice

- TimestampLastPrice
- NetChange
- Volume
- UnitofPrices
- UnitofVolumes

NOTE:

The fields BestBidPrice, BestAskPrice, BestBidVolume, BestAskVolume return the data for the latest available Bids and Asks for the specified symbol at the time when the API request is initiated.

Usage and Return Sample

https://api1.datasources.eex-group.com/getQuote/json?product=%23E.TTF_GND1

```
{ "results": [
  { "result": [
    {
      "Product": "#E.TTF_GND1",
      "Root": "GND1",
      "BestBidPrice": "5,025",
      "BestBidVolume": "10",
      "BestAskPrice": "",
      "BestAskVolume": "",
      "open": "14,8",
      "High": "15,2",
      "Low": "14,4",
      "close": "14,513",
      "Last": "15,025",
      "NetChange": "0",
      "Volume": 55945,
      "UnitofPrices": "EUR",
      "UnitofVolumes": "MWh"
    }
  ]}
]}
```

5.8.10 getIntraday

getIntraday returns intraday pricing over a range of time in a single day for a list of products and items for Bids, Asks and Trades.

The API will return a separate block of data for each event (Bid, Ask, Trade) that occurred in the specified period. Not every field will be returned for every event. See the “Available fields” section below for details.

NOTE: Gas Spot data is currently not available via getIntraday.

Parameters

Name	Optional	Description
Root	Yes	Identifier or a comma separated list of identifiers for the products required. Either a list of root or a list of products must be supplied
Products	Yes	Identifier or a comma separated list of identifiers for the products required
Return Format	Yes	If no return format is added, the default return format is XML JSON - return format is json XML – return format is xml CSV – return format is csv
Fields	Yes	Fields are dependent on the type of instrument to be retrieved, see the fields available in functions detailed in 5.6.1.2 to 5.6.1.14. This only works for products that trade throughout the day

Available Fields

- Root
- Product
- LongName
- MarketArea
- DeliveryStart
- DeliveryEnd
- Timestamp
- Price – For **Trade** events only
- Volume – For **Trade** events only
- NumberofTrades – For **Trade** events only
- BestBidPrice – For **Bid** events only
- BestBidVolume – For **Bid** events only
- BestAskPrice – For **Ask** events only
- BestAskVolume – For **Ask** events only
- UnitofPrices
- UnitofVolumes

Usage and Return Sample

<https://api1.datasources.eex-group.com/getIntraday/xml?root=DEBY&start=2023-04-05%2008:00&end=2023-04-05%2011:00>

```
<Result>
<Row>
<Field Name="Product">/E.DEBYF24</Field>
<Field Name="LongName">EEX German Power Base Year Future</Field>
<Field Name="Root">DEBY</Field>
<Field Name="BestBidPrice">149,800</Field>
<Field Name="BestBidVolume">1</Field>
<Field Name="BestAskPrice">151,400</Field>
<Field Name="BestAskVolume">1</Field>
<Field Name="MarketArea"/>
<Field Name="DeliveryStart">2024-01-01</Field>
<Field Name="DeliveryEnd">2024-12-31</Field>
<Field Name="TimeStamp">2023-04-05T08:00:06Z</Field>
<Field Name="Price"/>
<Field Name="Volume"/>
<Field Name="NumberofTrades"/>
<Field Name="UnitofPrices">EUR</Field>
<Field Name="UnitofVolumes">MWh</Field>
</Row>...
```

6. API Samples for Java and C#

6.1 Java Example Request

Below is the same example request written using Java.

```
package com.eex.test;

import java.io.DataOutputStream;
import java.io.IOException;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.util.Scanner;

import sun.misc.BASE64Encoder;

public class EEXRequest {

    // Main method
    public static void main(String[] args) {

        EEXRequest client = new EEXRequest();
        if (args.length >= 3) {
            try {
                client.execute(args[0], args[1], args[2]);
            } catch (Exception e) {
                e.printStackTrace();
            }
        } else {
            System.out.println("Usage : EEXRequest <URL> <user> <password>");
            // Example https://api1.datasource.eex-
group.com/getIntraday/xml?root=DEBM&start=2019-11-11%2015:55&end=2019-11-
11%2016:00 user1 password123
        }
    }

    public void execute(String serverURL, String user, String password) throws
MalformedURLException, IOException {

        // If you need to go through proxy by setting system properties

        // Proxy server
// System.getProperties().setProperty("proxySet", "true");
// System.getProperties().setProperty("proxyHost", proxy);
// System.getProperties().setProperty("proxyPort", proxyPort);

        // Proxy server credentials if needed
    }
}
```

```

//
    System.getProperties().setProperty("proxy.authentication.username",
proxyUsername);
//
    System.getProperties().setProperty("proxy.authentication.password",
proxyPassword);

        // Construct request
        URL httpFile = new URL(serverURL);
        HttpURLConnection connection =
(HttpURLConnection)httpFile.openConnection();
        connection.setRequestMethod("GET");
        connection.setRequestProperty("Content-Type", "application/x-www-form-
urlencoded");

        // Add credential to the request.
        BASE64Encoder enc = new sun.misc.BASE64Encoder();
        String encodedUsernameAndPassword = enc.encode((user + ":" +
password).getBytes());
        connection.setRequestProperty("Authorization", "Basic " +
encodedUsernameAndPassword);

        //Send request
        connection.setUseCaches(false);
        connection.setDoInput(true);
        connection.setDoOutput(true);
        DataOutputStream wr = new
DataOutputStream(connection.getOutputStream());
        wr.flush();
        wr.close();

        // Read response and write to screen
        Scanner scanner = new Scanner(connection.getInputStream());
        String responseBody = scanner.useDelimiter("\\A").next();
        System.out.println(responseBody);

    }

}

```

6.2 C# Example Request

Below is an example using C# to interact with the API.

```
using System;
using System.IO;
using System.Net;
using System.Net.Http;
using CsvHelper;
using CsvHelper.Configuration;

public class Event {
    public string Symbol { get; set; }
    public string LineType { get; set; }
    public string Country { get; set; }
    public string CompanyID { get; set; }
    public string ProdConsID { get; set; }
    public string UnitID { get; set; }
    public string ControlArea { get; set; }
    public string Source { get; set; }
    public string Type { get; set; }
    public string EventID { get; set; }
    public DateTime NUMStartDate { get; set; }
    public DateTime NUMEndDate { get; set; }
    public string NUMCapacity { get; set; }
    public string NonavailabilityReason { get; set; }
    public string Remarks { get; set; }
    public DateTime TimeStamp { get; set; }
    public string Status { get; set; }
    public DateTime PublicationTimeStamp { get; set; }
    public DateTime ModificationTimeStamp { get; set; }
    public string Commodity { get; set; }
    public string Facility { get; set; }

    public override string ToString() {
        return base.ToString();
    }
}

class Program {
    static void Main(string[] args) {
        var client = new HttpClient(new HttpClientHandler { Credentials = new
NetworkCredential("user", "password") });
        var resp = client.GetAsync("https://api1.datasources.eex-
group.com/getIntraday/xml?root=DEBM&start=2019-11-11%2015:55&end=2019-11-
11%2016:00").Result;
        if (!resp.IsSuccessStatusCode) {
            Console.Error.WriteLine(resp.StatusCode + " : " +
resp.ReasonPhrase);
            Environment.Exit(1);
        }
    }
}
```

```

    }

    using (var csv = new CsvReader(new
StreamReader(resp.Content.ReadAsStreamAsync().Result), new Configuration {
ShouldSkipRecord = arr => arr[0] == "Error" })) {
        csv.Read();
        csv.ReadHeader();
        while (csv.Read()) {
            var record = csv.GetRecord<Event>();
            Console.WriteLine(record);
        }
    }
}
}
}

```

The same rules apply to the parameters– Basic Example.

6.3 API Responses

API responses are available in three formats:

- a) JSON
- b) XML
- c) CSV

Example responses for each type to be found in the following section.

6.3.1 JSON sample return

<https://api1.datasources.eex-group.com/getIntraday/json?root=DEBM&start=2019-11-11%2015:55&end=2019-11-11%2016:00>

```

{ "results": [
  { "result": [
    {
      "Root": "DEBM",
      "Product": "\/E.DEBMF20",
      "LongName": "German Base Month Future",
      "MarketArea": "",
      "BestBidPrice": "45,6",
      "BestBidVolume": "5",
      "BestAskPrice": "45,8",
      "BestAskVolume": "10",
      "DeliveryStart": "2020-01-01",
      "DeliveryEnd": "2020-01-31",
      "TradeTimeStamp": "2019-11-11T15:56:02Z",
      "Price": "45,7",
      "Volume": 7440,
      "NumberOfTrades": 10,
      "UnitofPrices": "EUR",
      "UnitofVolumes": "MWh"
    }
  ]
}

```

```

},
{
  "Root": "DEBM",
  "Product": "\/E.DEBMF20",
  "LongName": "German Base Month Future",
  "MarketArea": "",
  "BestBidPrice": "45,6",
  "BestBidVolume": "5",
  "BestAskPrice": "45,8",
  "BestAskVolume": "10",
  "DeliveryStart": "2020-01-01",
  "DeliveryEnd": "2020-01-31",
  "TradeTimeStamp": "2019-11-11T15:56:15Z",
  "Price": "45,7",
  "Volume": 3720,
  "NumberofTrades": 5,
  "UnitofPrices": "EUR",
  "UnitofVolumes": "MWh"
}
]]
]]

```

This is one example record from the retrieved response. It is standard JSON containing name/value pairs for the returned data.

6.3.2 XML sample return

<https://api1.datasource.eex-group.com/getIntraday/xml?root=DEBM&start=2019-11-11%2015:55&end=2019-11-11%2016:00>

```

<Response status="OK">
<Item symbol="/E.DEBMH19" date="2/13/2019" time="12:00:00 PM"
open="42" high="42.2" low="41.75" volume="641" settle="42.87" last="42.05" ope
ninterest="135504"/>
</Response>

```

This is standard XML representing the fields returned in the response.

6.3.3 CSV sample return

<https://api1.datasource.eex-group.com/getIntraday/csv?root=DEBM&start=2019-11-11%2015:55&end=2019-11-11%2016:00>

```

Product,LongName,Root,BestBidPrice,BestBidVolume,BestAskPrice,BestAskVolume,Market
Area,DeliveryStart,DeliveryEnd,TimeStamp,Price,Volume,NumberofTrades,UnitofPrices,
UnitofVolumes
"/E.DEBMF20","EEX German Power Base Month Future","DEBM",,,,,,"2020-01-01","2020-
01-31","2019-11-11T15:56:02Z","45,70","7440","10","EUR","MWh"
"/E.DEBMF20","EEX German Power Base Month Future","DEBM",,,,,,"2020-01-01","2020-
01-31","2019-11-11T15:56:15Z","45,70","3720","5","EUR","MWh"

```

7. Good to Know

This chapter includes information that is good to know and/or only concerns specific requests or data types. These tips are not included under the respective chapters to not crowd the overall User Guide with special cases. Smaller known issues are also mentioned.

7.1 Agricultural Indices

The data field “ReferenceWeek” is empty and “LongName” is faulty prior to 22.11.2018.

7.2 Directly addressing Daily and Monthly EGIX via getHistory

EGIX can be addressed via getHistory. You cannot directly address all EGIX data via getHistory, but must differentiate between the daily and monthly EGIX:

```
https://api1.datasources.eex-group.com/getHistory/json?root=EGIX-D&producttype=index&start=2019-01-01&end=2020-12-31
```

```
https://api1.datasources.eex-group.com/getHistory/json?root=EGIX-M&producttype=index&start=2019-01-01&end=2020-12-31
```

7.3 Market area CZ VTP / OTE

For getSpot, the market area CZ VTP shows OTE. When filtering by market area, you must use OTE for getSpot:

```
https://api1.datasources.eex-group.com/getSpot/xml?returnType=results&commodity=NATGAS&marketArea=OTE&tradeDate=2021-01-04
```

For gas Derivatives, the market area is CZ:

```
https://api1.datasources.eex-group.com/getDerivatives?commodity=natgas&marketarea=CZ&producttype=futures&tradedate=2024-01-29&returntype=trades
```

7.4 New contracts first availability

Only roots and maturities that are tradable or have been tradable in the past are included in getPermission and getProduct.

Example:

EEX releases contracts for a new market area XX. The daily contracts XX01-XX31 all become tradable within the first month. Their master data is available in getPermission and getProduct as soon as they first get tradable. The contracts XX32-XX35 are placeholder contracts and are only used for special

cases. Especially for the contracts XX33-35, it can take many months or years until they first become tradable.

There is no disadvantage for the customer. Just be aware that, if you export the “getPermission” reply and address all received roots, the placeholder contracts may not be included. If they become tradable for the first time, you must add the root to your request list. You can already request these roots before they become tradable for the first time. You will not receive an error but only an empty reply for the specific roots.

7.5 No Gas Spot Data in getIntraday

Gas Spot data is currently not included for getIntraday. You can address intraday and eod data via getSpot. Best bid best ask data is not available for gas spot.

7.6 DeliveryStart/End and TradingStart/End for Derivatives are wrong

We are actively working on correcting our timestamps. In the meantime, please use our contract details file.

7.7 Unit of Volumes for Japanese Power Futures

The data field "UnitofVolumes" shows MWh for Japanese Power Futures. This is not inherently wrong, as "LotSize_Exch" shows the amount of MWh in a traded lot size. However, the price given under "SettlementPrice" is in Y/kwh and not in Y/MWh.

7.8 Difference of “old and new” EGSi and EGSi Futures

You can find the “old” EGSi (until 31st Dec 2022) via getSpot: <https://api1.datasource.eex-group.com/getSpot/xml?returnType=results&commodity=NATGAS&tradeDate=2021-01-04&root=EGSI>

The “new” EGSi is available via getIndex: <https://api1.datasource.eex-group.com/getIndex/xml?indexType=EGSI&DeliveryDate=2023-10-11&MarketArea=TTF>

The EGSi Futures are Derivatives and are, like all other Derivatives, available via getDerivatives.

7.9 Uniqueness of trade ID

The field TradeID represents the information from the CEF field "TICK_ID". The same trade ID can be shared for multiple legs of an implicit spread. Example:

<https://api1.datasources.eex-group.com/getDerivatives?commodity=power&producttype=futures&tradedate=2023-12-14&returntype=trades>

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Product	LongNa	Root	Maturit	Maturit	Market	Type	Underly	Strike	Deliver	Deliver	TradeTi	TradeID	Valid	Price
2	/E.DEBMF	EEX Germ	DEBM	2024-01	MONTH		EXCHANGE			2024-01-0	2024-01-3	2023-12-1	16419840	Yes	85.55
119	/E.F7BMF	EEX Frencl	F7BM	2024-01	MONTH		EXCHANGE			2024-01-0	2024-01-3	2023-12-1	16419840	Yes	83.5

If two or more trades share the same trade ID, have the same trading timestamp and the same traded volume, they are part of an implicit spread trade. Within our market data product, we only offer explicit trades for gas spot contracts. These spreads are marked as spreads, e.g.: EEX THE/TTF Natural Gas Day Spot, with the symbol #E.THE_GND1/TTF_GND1

7.9.1 Price Reporting for Implicit Trades

Please be advised of an update regarding the price information for implicit spread trades within the API data streams for derivatives.

Effective from the trading day of October 13th, 2025, the Price field for the individual legs of an implicit spread trade will be returned empty.

The volume information for these trades will continue to be provided.

This change applies to all futures and options derivatives. It does not impact index symbols, transparency data, or gas spot symbols. All historical trade data for dates prior to October 13th, 2025, remains unchanged.

7.9.2 Impact on getDerivatives with returnType=results

When requesting aggregated daily data (returnType=results), it is important to understand how implicit trades are handled:

OHLL Prices: Fields such as OpenPrice, HighPrice, LowPrice, and LastPrice are calculated exclusively from exchange trades that contain a price. Since the legs of implicit trades have an empty price field from October 13th, 2025 onwards, they are not considered in the calculation of OHLL values. If a product only has implicit trades on a specific day, no OHLC data is expected.

Volume and Trade Count: The volume from these trades is included in the aggregated volume and trade count fields.

7.10 Best bid best ask availability

There is no best bid best ask information available prior to May 24th 2021.

7.11 Emission Spot availability in getHistory

Emission spot data is available in getHistory under ProductType=Futures, not Spot:

<https://api1.datasource.eex-group.com/getHistory/xml?root=SEME,SEMA&ProductType=FUTURES&daysBack=1>

7.12 No Trade Registrations in getIntraday

The request is designed to follow the best bid best ask developments and view the resulting exchange trades. As the trade registrations have no connection to the best bid best ask developments, they are not included. Trade registrations are included in getHistory and getDerivatives or getSpot, depending on which contract type is requested.

8. Appendices

Market Area	Market Area Code
Central European Gas Hub Virtual Trading Point (Austria)	CEGH VTP
Czech Republic Virtual Trading Point	CZ VTP / OTE
Denmark	ETF
National Balancing Point (United Kingdom)	NBP
France	PEG
Punto di Scambio Virtuale (Italy)	PSV
Spain	PVB
Trading Hub Europe (Germany)	THE
Title Transfer Facility (Netherlands)	TTF
Zeebrugge Hub (Belgium) – merged with ZTP on Oktober 1 st 2023	ZEE
Zeebrugge Trading Point (Belgium)	ZTP