

part of eex group



ETS API Terms of Reference

04.01.2023
Paris

Rel. V6.10

Table of Contents

1.1	Reference	3
1.2	Audience	3
1.3	Purpose	3
1.4	Obligation to cooperate	3
1.5	Changes History Table	3
1.	Requirements	4
1.1	Testing	4
1.2	Request	4
1.2.1	Login and logout	4
1.2.2	Retrieve information request	5
2.2.2.1	Market results retrieval (and Trade Report retrieval)	5
2.2.2.2	Trading Limits inquiry requests	7
2.2.2.3	Assignments inquiry requests	8
2.2.2.4	Order inquiry requests	8
1.2.3	Message Validation	8

1. Introduction

1.1 Reference

This document is an amendment to the “API Software Design Guide” document.

1.2 Audience

This document is addressed to developers developing applications for the Public API messaging interface of ETS.

1.3 Purpose

The purpose of this document is to describe the expected behavior of the application using the API Public message interface of ETS. It is a set of requirements, rules and regulations which need to be adhered and applied to using the ETS API. These rules will ensure the proper functioning between client, systems and application on exchange member side or non-market participant side (e.g. Data Vendors) and ETS on the side of EPEX SPOT.

1.4 Obligation to cooperate

The rules mentioned in this document should be respected by the application’s developers. EPEX SPOT reserves the right to block access to the ETS API server and disconnect the customer’s API user from the EPEX SPOT system (API server and ETS server) at any time if the application of the member endangers the stability of the Trading System.

1.5 Changes History Table

Date	Version	Change description
16/12/2020	V6.4	Market results retrieval for Poland: part of the MRC auction
06/05/2021	V6.5	Interim Coupling Project: MRC Preliminary Results Publication time moves from 12:42 to 12:45 CET/CEST.
07/06/2021	V6.6	30 call FR Intraday auction time change on 30 June 2021 (1 st trading day): <ul style="list-style-type: none"> Order book closure will shift from 14:30 to 17:00 CET Auction results will be published 15 minutes after order book closure at 17:15
13/07/2021	V6.7	CH-IT IDA time changes
18/08/2021	V6.8	CH IDA auction names changes
17/02/2021	V6.9	Market results workaround pre 3.3.2 removal
04/01/2023	V6.10	Update of Max cancellation time for GB-IDA1 and GB-IDA2

1. Requirements

1.1 Testing

It is requested to the application provider to perform sufficient tests in the Simulation environment covering all functional features of its application before asking EPEX SPOT to release its application to production environment. The execution of specific tests case in Simulation can be requested by EPEX SPOT to members for quality assurance reasons.

1.2 Request

A client API application cannot run polls that keep the server busy continuously. The different policies below (e.g. login, market results retrieval) indicate you the rules your application needs to implement.

The client must use its own credentials (every customer should have its own certificates issued by or through EPEX SPOT) and no undocumented methods should be used.

Please note that only the synchronous mode is supported.

1.2.1 Login and logout

The previous version of this document only indicated this statement:

The client should not take up unnecessary resources. In particular login request should not be initiated for single requests.

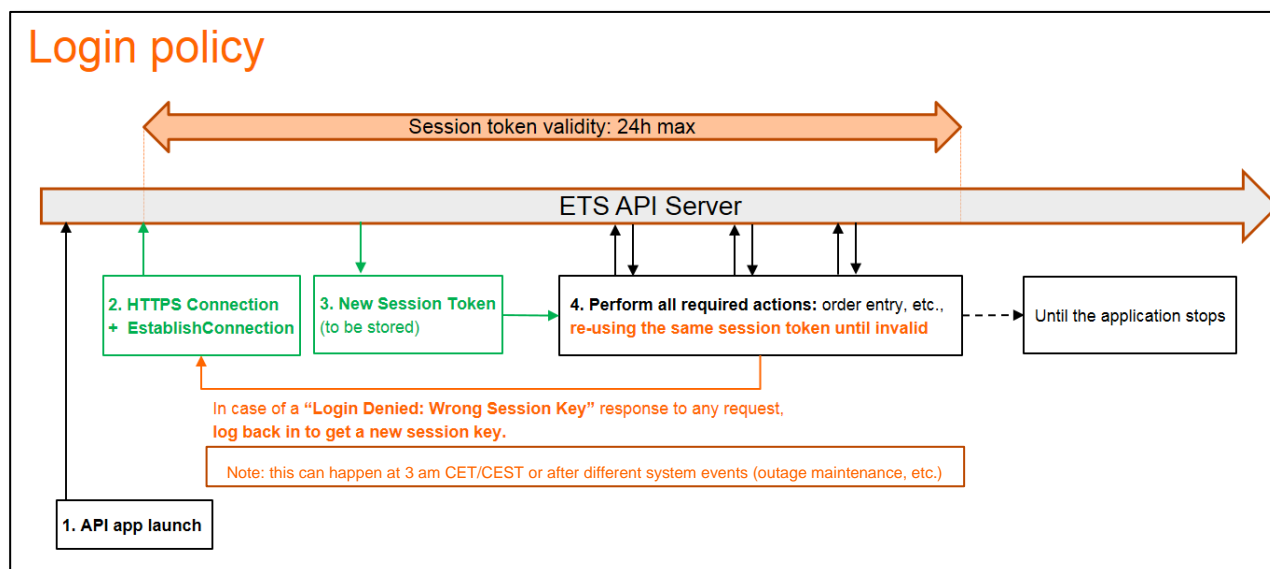
Considering the important growth of API applications in 2018, we had to go one step further.

Indeed:

- EstablishConnection (Login) is the most demanding API method, its use should be minimized,
- the Session Token (User Login Name + Session Key) delivered in the Establish Connection response remains valid even if the HTTPS connection has been closed because of inactivity: it is not required to re-log in to get a new token.

As a result, we ask you to follow the below connectivity cycle:

1. Your API application gets started
2. Log in: create an HTTP connection and send an EstablishConnection method (user + password)
3. Store the responded Session Token (user + session key)
4. Perform all required further actions by sending requests re-using the same token, until invalid
5. Once invalid, any request using it receives the response: **"Login Denied: Wrong Session Key"**
 - This will happen every early morning at 01:00 UTC when all connections to API servers are automatically kicked out for security purposes. At this moment all session keys become invalid.
 - This can also happen in case of an ETS maintenance or of an ETS outage.
6. When this happens go back to step 2 to get a new session key for the same user



HTTPS connections management:

- In case of an HTTP connection loss, just create a new one before sending a new request with the same session token.
- Please reserve the use of the *Keep Alive* method only for testing purposes, to check that your application can communicate the ETS API server (which does not require to be logged in), rather than for maintaining the HTTP connection alive.

1.2.2 Retrieve information request

2.2.2.1 Market results retrieval (and Trade Report retrieval)

Market results must be retrieved **automatically** only when relevant:

- Starting at the theoretical auction publication time
- Stopping once the auction results are *Final* (and thus cannot be cancelled anymore)

The shortest time interval between 2 Market result retrieval requests (*RetrieveMarketResultsFor* or *RetrieveTradesReportFor*) for the same area **should be 30 seconds**.

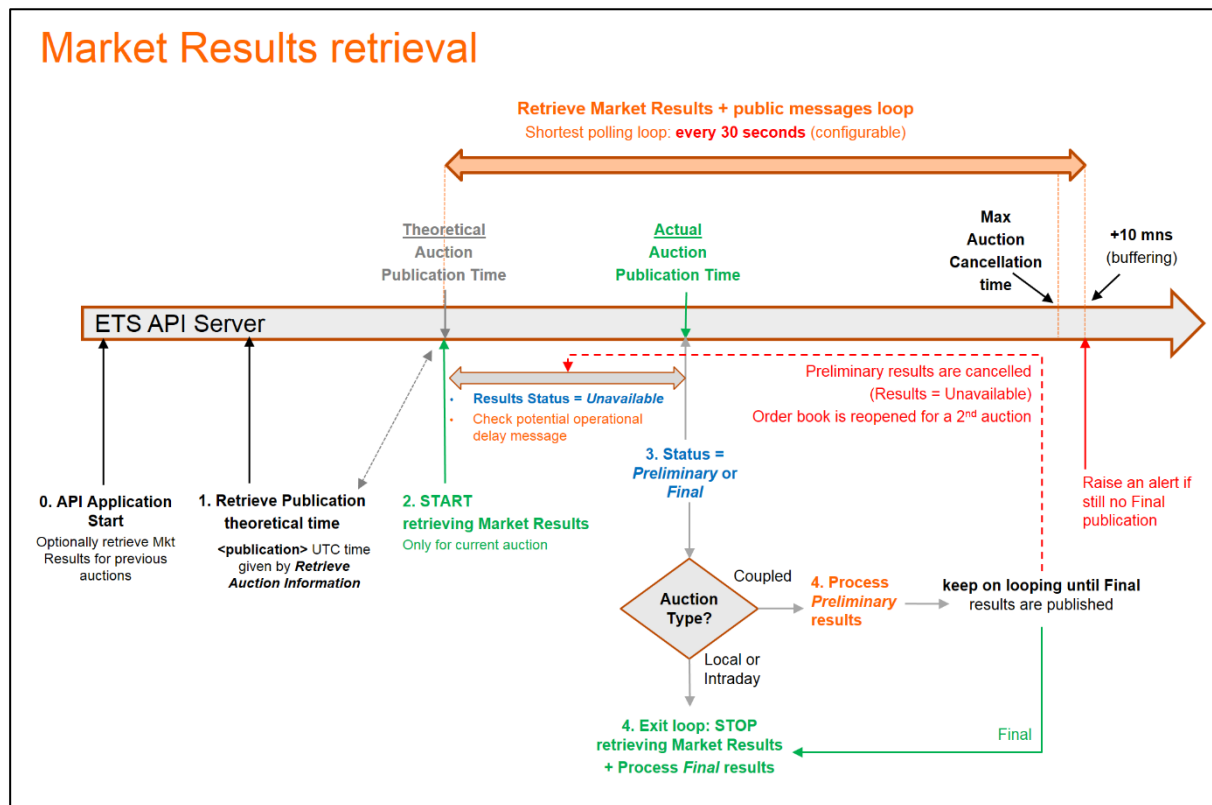
In addition: In order to be able to handle any exceptional situation, it is mandatory that your API application is able to retrieve Market Results **on demand/manually, at any time**. Any deviation from the standard publication process would be explicitly mentioned by EPEX in ETS messages (ETS client and ETS API) or by email.

Determining when to stop retrieving market results requires a good understanding of the different possible sequences of market results statuses and what is available via the API.

The sequence of market results statuses depends on the nature of the auction:

- Local and intraday auction:
 - *Unavailable* (Not yet published) -> *Final*.
 - There is no cancellation possibility.
- Day-ahead coupled auctions: *Unavailable* -> *Preliminary* -> *Final*
 - Thus, this never happened in production it is process wise possible that preliminary market results get cancelled:
 - *Unavailable* -> *Preliminary* -> *Cancelled* -> *Unavailable* -> *Preliminary* -> *Final*

Here is the Market Results Retrieval logic that must be implemented:



Please note that as of the 3.4.x API schema the *Auction Max Cancellation Time* can be retrieved via the API [Retrieve Auction Information](#) method.

Please find below useful auction characteristics to implement these logics: (recent changes are highlighted)

Auction	Auction Technical Name (in API, FTP files)	Theoretical publication time (CET)	Status of the first results	Max Cancellation Time in operational procedures	Max Cancellation Time For API apps (+10 min margin)
MRC	PWR-MRC-D+1	12:45 (ICP Change)	Preliminary	15:30 20:00 for Nordics (in case of decoupling). (*)	15:40
MRC	PWR-MRC-D+1	GB: 10:30 since Brexit	Final	14:00	14:10
15 Call DE	PWR-15-call-DE-D+1	15: 20	Final	17:00	17:10
30 Call FR	PWR-30-call-FR-D+1	17:15	Final	19:30	19:40
15 Call AT	PWR-15-call-AT-D+1	15:20	Final	17:00	17:10
15 Call BE	PWR-15-call-BE-D+1	15:40	Final	17:00	17:10

15 Call NL	PWR-15-call-NL-D+1	15:40	Final	17:00	17:10
30 Call GB	PWR-30-call-GB-D+1	16:45	Final	17:30	17:40
CH	PWR-CH-D+1	11:10	Final	14:30	14:40
CH/IT-IDA1	PWR-CH-IDA1-D+1	17:55	Final	18:55	19:05
CH/IT-IDA2	PWR- CH-IDA2-D	10:45	Final	11:00	11:10
GB - IDA1	PWR-SEM—GB-D+1	19:00	Preliminary	19:45 GMT / 20:45 CET	19:55 GMT / 20:55 CET
GB - IDA2	PWR-SEM—GB-D	09:30	Preliminary	10:15 GMT / 11:15 CET	10:25 GMT / 11:25 CET

(*) **In case of Nordic decoupling:** a specific message (Nordic - ext_01), readable by API apps using the *RetrieveMessagesUnreadOnly* method, when received (before 15:40), should tell the API app to continue looping until 20:00 and stop when results are final:

"Nordic - ext_01

Dear Member,

As a consequence of the decoupling from MRC, the Nordic order books will be reopened at HH:MM for exactly 20 minutes.

After the closing of the order books, a Nordic Regional Coupling shall be performed and the Market Coupling Results are expected no later than 20:00."

Alert management in case no Final result is published (at the end of the automatic retrieval period):

In case the "Max Cancellation time + 10 mns" is reached without having retrieved *Final* results:

- please check instructions sent by EPEX via ETS messages (readable by API apps) or via email
- 2 scenarios can happen:
 - 1) if EPEX happens to publish market results later please use the manual/on demand market results retrieval requested above to force your application to request market results outside of the automatic retrieval period.
 - 2) in the extreme case where EPEX could not publish market results at all, EPEX would explain that reference prices are being calculated and published on the EPEX website. This is the only exception where applications are authorized to read market results straight from the website. No publication would be done via the ETS Client or the ETS API.

2.2.2.2 Trading Limits inquiry requests

Your API application is authorized to send the required set of TL requests to refresh its knowledge of all TL information at different moments:

- **periodically: every 3 seconds maximum** (meaning there should be at least 3 seconds between 2 sets of TL requests) ; **this periodicity must be configurable** (so that each API app can easily change it on request of EPEX SPOT if technically required, to remain flexible).
- **AND after each order management action** (if the API app needs a refresh before the next periodic TL refresh)
- **AND after each Market Results publication** (if the API app needs a refresh before the next periodic TL refresh)
- **AND "on demand"** (not periodic), for instance:
 - For API apps with a GUI:
 - To populate a TL screen (initial data retrieval/display)
 - On demand of an end user (e.g. potential forced manual TL screen refresh)

1. RetrieveCentralCounterPartyAndSettlementMemberNames

2. RetrieveTradingLimits
3. RetrieveTradingLimitExtendedDetails

2.2.2.3 Assignments inquiry requests

As the assignments (e.g. portfolios, delivery areas) of each API user will not change all the time, “retrieve information” requests must only be sent:

- after the start of the client application in order to retrieve the initial assignments,
- and then at a frequency that that does not keep the server continuously busy, to remain aware of potential new assignments.

1. RetrieveTradableAreaPortfolioInformations
2. RetrieveTradableAreaSets
3. RetrieveTradableAreas
4. RetrieveTradablePortfolios
5. RetrieveViewableAreaPortfolioInformations
6. RetrieveViewableAreas
7. RetrieveViewablePortfolios

2.2.2.4 Order inquiry requests

API Applications may need to retrieve orders in order to:

- 1) **Build a consolidated view of all orders submitted for a given auction**
 - o For the next coming auction: such requests should be sent at a frequency that that does not keep the server continuously busy.
 - o For past and further delivery days: the request frequency should be lower as for the next coming auction.
- 2) **Check whether or not an order management request has been taken into account by ETS:**
 - o It may happen that when the ETS system is busy your application receives the following error after an order action:
 - ErrorId = “OA 012”, Error text = “Trading System did not answer within: [x] Seconds”
 - o In such an case it is likely that your request was processed but it is customers responsibility to check whether this is the case or not : your API application must send a Retrieve order request to check if the entry, the update or cancellation has been taken into account.
 - o In this context the request can be sent immediately after having received the error needing to retrieve orders

If this “server busy” error occurs when your application processes a loop of several orders actions, two implementation strategies are possible, with equal efficiency:

- a) You handle the error immediately before continuing the loop with the next order action
- b) Or you temporarily ignore the error and finish the loop to process remaining order actions, and only check the error afterwards at the end of the loop.

1.2.3 Message Validation

In general, ETS API will validate messages. Invalid messages will be rejected. If an application is erroneous and sends invalid messages in short frequency, this could have the effect of a denial of service attack and can result in a suspension of the related application.

Client applications ensure that messages sent are valid according to the specified XML schema (WSDL). Messages sent with an invalid XML schema that cannot be processed by API server will be rejected using a native error response message (utf-8 message).

All mandatory message properties have to be set when sending a request to API server. ETS API server will reject messages with insufficient message properties.