



# SSE Manual ETPA

## Introduction

While the API can be very helpful to get all the information for a certain period it will not help you to keep up to date with all the trades and orders. The amount of data that is transmitted to our participant is simply too large and we would not recommend keeping using the API to stay up to date with the orders and trades, since that would create too much load on our machines. To solve this problem, we have come up with a solution called SSE. With SSE you will receive constant update on all orders and trades.



## Table of Contents

<b>Introduction</b> .....	<b>2</b>
<b>1. Connecting to the SSE</b> .....	<b>4</b>
1.1 Listening to messages .....	4
<b>2. Channels</b> .....	<b>5</b>
2.1 Announcements .....	5
2.2 Orderbook.....	5
2.2.1 Intraday Orderbook.....	6
2.2.2 Ex-post Orderbook .....	7
2.2.3 GOPACS Orderbook.....	8
2.2.4 XBID Orderbook .....	9
2.3 Trades.....	12
2.3.1 Intraday Trades.....	12
2.3.2 Ex-post Trades .....	13
2.3.3 GOPACS Trades.....	14
2.3.4 XBID Trades .....	14
2.3.5 XBID Public Trades.....	15

## 1. Connecting to the SSE

For connecting to the SSE, you need to identify yourself with the API Key. This API Key is placed in the header of the request. Both acceptance and production are different environments, so when changing environments, you need to change both the URL and API Key. If you have trouble accessing the SSE endpoint, please send an e-mail to [support@etpa.nl](mailto:support@etpa.nl) for an example script of connecting to the SSE channels.

- Endpoint URL: `https://[acc-]trading.etpa.nl/public-sse/*`
- API header: `api_key`
- Useful link: <https://javascript.info/server-sent-events>

*Example to connect:*

```
var option = {
  headers: {
    api_key: "api_key"
  }
};

const sse = new EventSource(url, option);
```

When an invalid api key is provided a 403 error will be thrown

### 1.1 Listening to messages

When you are subscribed to a channel you will get 4 values back

Type: This is the type of the message. This can be either message or

Data: The data of the channel

lastEventId: The id of the event. This is currently only used for the XBID orderbook channel.

origin: The URL.

Listening to messages:

```
sse.onmessage = (event) => {
  console.log(event);
}
```

## 2. Channels

At ETPA we provide all different kind of channels. This chapter describes all the different channels we provide.

### 2.1 Announcements

URL: /announcements

For the announcements we have set an initial state to retrieve all the current ACTIVE announcements.

For the announcement channel you will receive a message for every newly created or deleted and/or expired announcement in the application. To make a distinction between created and deleted announcements, the “active” field from the message can be used.

*New announcement example:*

```
[{
  "id": "147f6be5-a0bf-461c-9423-08988e1f47a5",
  "message": "Test announcement",
  "created": 1660913263900,
  "start": 1660913252399,
  "end": 1661518052399,
  "active": true
}]
```

*Deleted announcement example:*

```
[{
  "id": "147f6be5-a0bf-461c-9423-08988e1f47a5",
  "message": null,
  "created": null,
  "start": null,
  "end": null,
  "active": false
}]
```

### 2.2 Orderbook

For the orderbook channels (ex-post, intraday, XBID and GOPACS) we have set an initial state. The initial state will send you all the orders which are currently in the orderbook. The initial state message will be one message containing all the orders which are currently in the orderbook. After the initial state is send it will automatically send you all the new updates of the orderbook.

The differentiation between newly created and deleted orders can be done by using the field: 'type'.

- INFO: The order has been created
- WARNING: The order has been cancelled
- REFRESH: The order has been updated due to partial match

**NOTE: Updating an order will result in one order deleted (WARNING) and one order created message (INFO). When the order is updated the orderId changes. You can keep track of the order with the front-end ID.**

You will automatically receive all the necessary information. Some fields will be automatically masked. These fields are:

- participantId
- ean
- allowedToBeUsedForIdcons
- individualFullName
- individualId
- metadata.

### 2.2.1 Intraday Orderbook

URL: /intraday-orderbook

*New intraday order example:*

```
[
  {
    id: 'd369b9cd-dc25-4a8f-b13f-82d9fb97d742',
    type: 'INFO',
    order: {
      id: 'd369b9cd-dc25-4a8f-b13f-82d9fb97d742',
      frontendId: '0830d163-e74c-4a6d-aff7-1b308618bc05',
      price: 1,
      quantity: 1,
      product: 'ELECTRICITY',
      timeblock: 'INTRADAY',
      type: 'BUY',
      start: 1679050800000,
      end: 1679054400000,
      participantId: 'd348e1d2-64e6-49c4-97b7-10f863c8c1b3',
      created: 1679045864264,
      priority: 1679045864264,
      ean: '',
      allowedToBeUsedForIdcons: false,
      individualFullName: 'Daniël Otter',
      individualId: 'd01a03ad-caa1-4500-807e-e1ee337fba5a',
      originalQuantity: 1,
      customExpirationTime: 1679049900000,
      metadata: null
    }
  }
]
```

*Delete intraday order example:*

```
[
  {
    id: 'd369b9cd-dc25-4a8f-b13f-82d9fb97d742',
```

```

type: 'WARNING',
order: {
  id: 'd369b9cd-dc25-4a8f-b13f-82d9fb97d742',
  frontendId: '0830d163-e74c-4a6d-aff7-1b308618bc05',
  price: 1,
  quantity: 1,
  product: 'ELECTRICITY',
  timeblock: 'INTRADAY',
  type: 'BUY',
  start: 1679050800000,
  end: 1679054400000,
  participantId: 'd348e1d2-64e6-49c4-97b7-10f863c8c1b3',
  created: 1679045864264,
  priority: 1679045864264,
  ean: '',
  allowedToBeUsedForIdcons: false,
  individualFullName: 'Daniël Otter',
  individualId: 'd01a03ad-caa1-4500-807e-e1ee337fba5a',
  originalQuantity: 1,
  customExpirationTime: 1679049900000,
  metadata: null
}
}
]

```

## 2.2.2 Ex-post Orderbook

URL: /expost-orderbook

This channel can only be accessed when the participant has ex-post rights enabled.

*New ex-post order example:*

```

[
  {
    id: '66455162-c497-4ca7-b869-bc330271ddee',
    type: 'INFO',
    order: {
      id: '66455162-c497-4ca7-b869-bc330271ddee',
      frontendId: '8f0587ba-26d4-41eb-8e7e-1ba410d24a4a',
      price: 1,
      quantity: 1,
      product: 'ELECTRICITY',
      timeblock: 'EXPOST',
      type: 'BUY',
      start: 1679041800000,
      end: 1679042700000,
      participantId: 'd348e1d2-64e6-49c4-97b7-10f863c8c1b3',
      created: 1679046210503,
      priority: 1679046210502,
      ean: null,
      allowedToBeUsedForIdcons: false,
      individualFullName: 'Daniël Otter',
      individualId: 'd01a03ad-caa1-4500-807e-e1ee337fba5a',
      originalQuantity: 1,
      customExpirationTime: null,
      metadata: null
    }
  }
]

```

```
]
```

### Ex-post order deleted message:

```
[
  {
    id: '66455162-c497-4ca7-b869-bc330271ddee',
    type: 'WARNING',
    order: {
      id: '66455162-c497-4ca7-b869-bc330271ddee',
      frontendId: '8f0587ba-26d4-41eb-8e7e-1ba410d24a4a',
      price: 1,
      quantity: 1,
      product: 'ELECTRICITY',
      timeblock: 'EXPOST',
      type: 'BUY',
      start: 1679041800000,
      end: 1679042700000,
      participantId: 'd348e1d2-64e6-49c4-97b7-10f863c8c1b3',
      created: 1679046210503,
      priority: 1679046210502,
      ean: null,
      allowedToBeUsedForIdcons: false,
      individualFullName: 'Daniël Otter',
      individualId: 'd01a03ad-caa1-4500-807e-e1ee337fba5a',
      originalQuantity: 1,
      customExpirationTime: null,
      metadata: null
    }
  }
]
```

### 2.2.3 GOPACS Orderbook

URL: /gopacs-orderbook

This channel can only be accessed when the participant has GOPACS rights enabled.

### New GOPACS order example:

```
[{
  "id": "4ce41e23-1b7e-4276-995a-c75f7c4b1855",
  "type": 'INFO',
  "order": {
    "id": "4ce41e23-1b7e-4276-995a-c75f7c4b1855",
    "frontendId": "da84cae7-e078-4bb3-a9b8-958fefdfa791",
    "price": 1,
    "quantity": 1,
    "product": "ELECTRICITY",
    "timeblock": "GOPACS",
    "type": "BUY",
    "start": " 1672149600000",
    "end": 1672153200000,
    "participantId": "b548d968-fc4b-4c87-a23e-6834156b6a36",
    "created": 1672148235169,
    "priority": 1672148235168,
    "ean": "111111111111111111",
    "allowedToBeUsedForIdcons": true,
    "individualFullName": "Owner of participant Zero",
    "individualId": "4b2ba585-f256-4d3a-8387-4315e805de5b",
```



```
    "originalQuantity": 1,
    "customExpirationTime": 1672148700000,
    "metadata": null
  }
}]
```

### Delete GOPACS order example:

```
[{
  "id": "4ce41e23-1b7e-4276-995a-c75f7c4b1855",
  "type": 'WARNING',
  "order": {
    "id": "4ce41e23-1b7e-4276-995a-c75f7c4b1855",
    "frontendId": "da84cae7-e078-4bb3-a9b8-958fefdfa791",
    "price": 1,
    "quantity": 1,
    "product": "ELECTRICITY",
    "timeblock": "GOPACS",
    "type": "BUY",
    "start": " 1672149600000",
    "end": 1672153200000,
    "participantId": "b548d968-fc4b-4c87-a23e-6834156b6a36",
    "created": 1672148235169,
    "priority": 1672148235168,
    "ean": "111111111111111111",
    "allowedToBeUsedForIdcons": true,
    "individualFullName": "Owner of participant Zero",
    "individualId": "4b2ba585-f256-4d3a-8387-4315e805de5b",
    "originalQuantity": 1,
    "customExpirationTime": 1672148700000,
    "metadata": null
  }
}]
```

## 2.2.4 XBID Orderbook

URL: /xbid-orderbook

This URL can only be accessed when you have the XBID functionality enabled.

For the XBID orderbook channel we have introduced some additional information to the message. Such as lastEventId and ping. This will allow you to keep track of the messages and make sure that you are always connected.

### 2.2.4.1 Last event id message

To retrieve the last event id messages, you need to listen to the last event Id message. The initial state will have the lastEventId 0 and with every message this number will be increased by 1.

#### Listening to the last event ids:

```
sse.onmessage = (event) => {
  console.log("last Event id: ", event.lastEventId);
}
```

To get the full message when you are subscribed to the XBID orderbook.

#### 2.2.4.2 Ping messages

For the XBID orderbook we have also introduced ping messages. This will let you know if you are still connected. The ping messages contain empty data. These messages will be send every 2 seconds. If our application notices you are not connected it will try to reconnect.

To retrieve the ping messages, you can do the following:

**Add listener for ping messages:**

```
sse.addEventListener("ping", (event) => {
  console.log("Ping: ", event.data);
});
```

#### 2.2.4.3 Orderbook message

To get the message from the orderbook data you need to look at the data object. This contains the order data.

**Example listening to data:**

```
sse.onmessage = (event) => {
  console.log(event.data);
}
```

The response for the XBID orderbook is slightly different than we have for ex-post, intraday and GOPACS. There are also some different action types which define which action has been taken.

- **CREATED:** The order has been created.
- **UPDATED:** The order has been updated.
- **DELETED:** The order has been deleted.
- **PARTIALLY\_MATCHED:** The order has been partially matched.
- **FULLY\_MATCHED:** The order has been fully matched.

**New XBID intraday order example:**

```
[{
  "action": "CREATED",
  "data": {
    "xbidOrderId": 18712282,
    "id": "efc4923e-a551-4f23-910b-da786b4d1520",
    "orderType": "BUY",
    "xbidOrderType": "REGULAR",
    "participantId": "dd7472d3-9ab4-4ed6-83ad-490ef4bd78a5",
    "individualId": "de0ff7c5-a9e6-45bf-aece-174ebb5ac973",
    "individualFullName": "Barend Otter",
    "capacity": 1,
    "originalCapacity": 1,
    "created": 1699540137728,
    "priority": null,
    "deliveryStartTime": 1699542000000,
    "deliveryEndTime": 1699545600000,
  }
}]
```

```

"unitPrice": 1,
"customExpirationTime": null,
"metadata": null,
"totalPrice": 1,
"orderDeliveryArea": {
  "country": "NL",
  "eic": "10YNL-----L",
  "name": "NL - TTN"
},
"orderExecution": "NON",
"orderStatus": "ACTIVE",
"localMarket": false,
"revisionNo": 1
}
}]

```

An update of the order can happen when XBID approves the creation of your order or when the order is updated by ETPA or due to partial match.

#### *Update of XBID intraday order example:*

```

[
{
  "action": "UPDATED",
  "data": {
    "xbidOrderId": 18712285,
    "id": "24493d81-9e80-45f6-8655-3bafb143c9e7",
    "orderType": "BUY",
    "xbidOrderType": "REGULAR",
    "participantId": "dd7472d3-9ab4-4ed6-83ad-490ef4bd78a5",
    "individualId": "de0ff7c5-a9e6-45bf-aece-174ebb5ac973",
    "individualFullName": "Barend Otter",
    "capacity": 2,
    "originalCapacity": 1,
    "created": 1699538668461,
    "priority": null,
    "deliveryStartTime": 1699556400000,
    "deliveryEndTime": 1699560000000,
    "unitPrice": 1,
    "customExpirationTime": null,
    "metadata": null,
    "totalPrice": 2,
    "orderDeliveryArea": {
      "country": "NL",
      "eic": "10YNL-----L",
      "name": "NL - TTN"
    },
    "orderExecution": "NON",
    "orderStatus": "ACTIVE",
    "localMarket": false,
    "revisionNo": 1
  }
}
]

```

#### *Delete of XBID intraday order example:*

```

{
  action: 'DELETED',
  data: {
    xbidOrderId: 18231740,
    id: '71f4f712-7f92-46be-8914-fdeaba3b87fc',
    orderType: 'BUY',

```

```
xbidOrderType: 'REGULAR',
participantId: 'd348e1d2-64e6-49c4-97b7-10f863c8c1b3',
individualId: 'd01a03ad-caa1-4500-807e-e1ee337fba5a',
capacity: 12,
originalCapacity: 12,
created: 1688464174817,
priority: 1688464174000,
deliveryStartTime: 1688477400000,
deliveryEndTime: 1688478300000,
unitPrice: 12,
customExpirationTime: null,
metadata: null,
totalPrice: 36,
orderDeliveryArea: 'NL',
marketAreaId: 'NL',
orderExecution: 'NON',
orderStatus: 'ACTIVE',
localMarket: false
}
}
```

## 2.3 Trades

At ETPA we provide different trade channels. This is for all the products we offer.

### 2.3.1 Intraday Trades

URL: /intraday-trades

For the intraday trades channels we also set an initial state. This is based on the start of the day (00:00). The initial state will be sent as one message in list. After this you will automatically retrieve all the new trades.

Some fields will be masked based on security reasons. These fields are:

Buyer-sensitive information

- buyerId
- buyerEan
- orderIdBuy
- buyOrderMetadata.

Seller-sensitive information:

- sellerId
- sellerEan
- orderIdSell
- sellOrderMetadata
- 

Congestion-sensitive information:

- congestionId
- comment.

*Example intraday trade:*

```
[{
  "id": "63ac7e4bf2e6cf733cf94d",
  "type": "INFO",
  "trade": {
    "id": "63ac7e4bf2e6cf733cf94d",
    "tradeId": "02e3c81a-96c3-42be-896c-d96ec3ba9ae0",
    "productType": "ELECTRICITY",
    "timeblock": "INTRADAY",
    "buyerId": "5fcb140c-624c-4499-b9f0-8441d51bf8a1",
    "sellerId": "",
    "buyerGridOperator": false,
    "sellerGridOperator": false,
    "orderIdBuy": "3ea7194e-e171-49e1-8bcb-bea05a407dc2",
    "orderIdSell": "",
    "buyerEan": "",
    "sellerEan": "",
    "buyOrderMetadata": null,
    "sellOrderMetadata": {},
    "quantity": 1,
    "start": 1672254000000,
    "end": 1672257600000,
    "executed": 1672248906346,
    "price": 1,
    "type": "intra-day",
    "comment": "",
    "congestionId": "",
    "duration": 1,
    "isCongestionTrade": false
  }
}]
```

### 2.3.2 Ex-post Trades

URL: /expost-trades

This channel can only be accessed when the participant has ex-post rights enabled.

*Example message ex-post trade:*

```
{
  id: '63d8f9264801b70bb77af8bd',
  type: 'INFO',
  trade: {
    id: '63d8f9264801b70bb77af8bd',
    tradeId: 'ada69bb6-9a01-4c5f-a061-287760472a65',
    productType: 'ELECTRICITY',
    timeblock: 'EXPOST',
    buyerId: 'd348e1d2-64e6-49c4-97b7-10f863c8c1b3',
    sellerId: '',
    buyerGridOperator: false,
    sellerGridOperator: false,
    orderIdBuy: '726a0371-2498-4c08-9c96-8d1a5653fff4',
    orderIdSell: '',
    buyerEan: null,
    sellerEan: '',
    buyOrderMetadata: {},
    sellOrderMetadata: {},
    quantity: 10,
    start: 1675135800000,
  }
}
```

```
end: 1675136700000,  
executed: 1675163942374,  
price: 14,  
type: 'ex-post',  
comment: '',  
congestionId: '',  
duration: 0.25,  
isCongestionTrade: false  
}  
}
```

### 2.3.3 GOPACS Trades

URL: /gopacs-trades

This channel can only be accessed when the participant has GOPACS rights enabled.

*Example message GOPACS trade:*

```
{  
  id: '63d8f9264801b70bb77af8bd',  
  type: 'INFO',  
  trade: {  
    id: '63d8f9264801b70bb77af8bd',  
    tradeId: 'ada69bb6-9a01-4c5f-a061-287760472a65',  
    productType: 'ELECTRICITY',  
    timeblock: 'GOPACS',  
    buyerId: 'd348e1d2-64e6-49c4-97b7-10f863c8c1b3',  
    sellerId: '',  
    buyerGridOperator: false,  
    sellerGridOperator: false,  
    orderIdBuy: '726a0371-2498-4c08-9c96-8d1a5653fff4',  
    orderIdSell: '',  
    buyerEan: null,  
    sellerEan: '',  
    buyOrderMetadata: {},  
    sellOrderMetadata: {},  
    quantity: 10,  
    start: 1675135800000,  
    end: 1675136700000,  
    executed: 1675163942374,  
    price: 14,  
    type: 'gopacs',  
    comment: '',  
    congestionId: '',  
    duration: 0.25,  
    isCongestionTrade: true  
  }  
}
```

### 2.3.4 XBID Trades

URL: /xbid-trades

The XBID trades contain all the trades where ETPA participants are part of the trades.

### Example message XBID trade

```
{
  "tradeId": "78485ecf-2b6b-4732-97ee-cb8c1e76fbbd",
  "xbidTradeId": 11153200,
  "tradeQuantity": 1,
  "mwhTraded": 1.00,
  "tradePrice": 1,
  "totalPrice": 1,
  "deliveryStartTime": 1700478000000,
  "deliveryEndTime": 1700481600000,
  "executionTime": 1700475383375,
  "buyOrderId": "df3152ce-b37e-41a3-9bc1-23828e867ba3",
  "sellOrderId": "",
  "buyParticipantId": "dd7472d3-9ab4-4ed6-83ad-490ef4bd78a5",
  "sellParticipantId": "",
  "buyIndividualId": "de0ff7c5-a9e6-45bf-aece-174ebb5ac973",
  "sellIndividualId": "",
  "buyIndividualFullName": "Barend Otter",
  "sellIndividualFullName": "",
  "buyDeliveryArea": {
    "country": "NL",
    "eic": "10YNL-----L",
    "name": "NL - TTN"
  },
  "sellDeliveryArea": {
    "country": "NL",
    "eic": "10YNL-----L",
    "name": "NL - TTN"
  },
  "reverted": false,
  "buyOrderMetadata": null,
  "sellOrderMetadata": null
}
```

### 2.3.5 XBID Public Trades

URL: /xbid-public-trades

The XBID public trades contain all the XBID public trades in the Dutch delivery area.

#### Example message XBID trade:

```
{
  "tradeId": "11153199",
  "xbidTradeId": 11153199,
  "revisionNo": 1,
  "tradeQuantity": 1.00,
  "mwhTraded": 1.0000,
  "tradePrice": 1.00,
  "totalPrice": 1.0000,
  "deliveryStartTime": 1700478000000,
  "deliveryEndTime": 1700481600000,
  "executionTime": 1700475317208,
  "buyDeliveryArea": {
    "country": "NL",
    "eic": "10YNL-----L",
    "name": "NL - TTN"
  },
  "sellDeliveryArea": {
```

```
"country": "NL",  
"eic": "10YNL-----L",  
"name": "NL - TTN"  
},  
"reverted": false  
}
```

