



Your Trades List v1

Document Revision 1.1
Date of Issue: 10 December 2020
Date of revision: 18 January 2021

Table of Contents

1. Purpose	3
2. Glossary of Terms.....	3
3. Technical Standards.....	3
4. Request Header	3
5. API Listing.....	4
5.1 Your Trades List service (POST method).....	4
6. Response Codes	10
6.1 Request validation error codes	10
6.2 HTTP Status codes.....	11

1. Purpose

To provide the API end point information and examples of the web services available for Your Trades List.

2. Glossary of Terms

Term	Meaning
LWIN	LWIN - the Liv-ex Wine Identification Number – serves as a universal wine identifier for the wine trade. LWIN is a unique seven to eighteen-digit numerical code that can be used to quickly and accurately identify a product. LWIN allows wine companies to keep their preferred naming system, while introducing a new universal code.
SIB	Standard in Bond trade terms: https://www.liv-ex.com/knowledge/liv-ex-trading-contracts/
SEP	Standard En Primeur: https://www.liv-ex.com/knowledge/liv-ex-trading-contracts/
Special (X)	Special contract trade terms: https://www.liv-ex.com/knowledge/liv-ex-trading-contracts/
Contract Type	Contract type is a generic term for SIB, SEP or Special (X).

3. Technical Standards

- Permitted users will be issued with a unique token (CLIENT_KEY) and password (CLIENT_SECRET) combination to control the access for all the web services covered under Exchange Integration.
- The web services will consume and produce both XML and JSON. The user can provide the content type in the request header. If the user does not provide any information, then the default content type will be JSON.
- The service supports ISO 8601.
- The service only support HTTPS protocol for client and server communications.
- The API will support the following methods:
 - POST for create operation
- Pretty printing for output readability only is supported if required
- Compression for bandwidth savings are used
- Authentication mechanism will be custom based on CLIENT_KEY and CLIENT_SECRET

4. Request Header

This information will be used to authenticate valid access to the REST API. Each user will have to provide the following information in the request header. Please note that the API expects the 4 headers as listed within this documentation and submitting a request with additional headers may lead to errors and/or failed responses.

Parameter

Name	Mandatory	Description
CLIENT_KEY	Y	A valid GUID which will be unique for each user.

CLIENT_SECRET	Y	Password/Secret for the merchants CLIENT_KEY.
ACCEPT	Y	Accept header is a way for a client to specify the media type of the response content it is expecting. The values for the content type will be application/json or application/xml. If no/ invalid content type is found in the request, then JSON format will be used by default.
CONTENT-TYPE	Y for POST requests	Content-type is a way to specify the media type of request being sent from the client to the server. The values for the content type will be application/json or application/xml. If no/ invalid content type is found in the request, then JSON format will be used by default.

Example header

```
CLIENT_KEY: 12A34BC56-DE7F-89G0-H1J2345K678L
CLIENT_SECRET: dummy_password
ACCEPT: application/json
CONTENT-TYPE: application/json
```

Invalid header (JSON response)

```
{
  "status": "Unauthorized",
  "statusCode": "401",
  "message": "Request was unsuccessful",
  "internalErrorCode": "R000",
  "apiInfo": {
    "version": "1.0",
    "timestamp": 1599660380849,
    "provider": "Liv-ex"
  }
}
```

Invalid header (XML response)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Response>
  <Status>Unauthorized</Status>
  <HttpCode>401</HttpCode>
  <Message>Request was unsuccessful</Message>
  <InternalErrorCode>R000</InternalErrorCode>
  <ApiInfo>
    <Version>1.0</Version>
    <Timestamp>2020-09-09T17:00:51.094+01:00</Timestamp>
    <Provider>Liv-ex</Provider>
  </ApiInfo>
</Response>
```

5. API Listing

5.1 Your Trades List service (POST method)

Description

This service can be used to request a list of Liv-ex trades.

There are no mandatory parameters. Calling the service with no request parameters specified will result in all trade events over the last 2 years being returned. Request attributes can be used to filter results as required.

Base URI

[accounts/v1/yourTradesList](https://api.liv-ex.com/accounts/v1/yourTradesList)

URI Pagination Parameters

Name	Mandatory	Description
limit	N default =50	Number of results to include per response. A "result" is counted as one Liv-ex trade. Values: '1' to '250' (maximum) Type: integer
offset	N default = 1	The # of the response page to return. Starting value always = 1. Type: integer

Example URI including pagination

<https://api.liv-ex.com/accounts/v1/yourTradesList?limit=25&offset=1>

Where a request returns more than a single page of results, consuming systems should iterate the offset value until all results have been received (a sequence of POSTs with the body remaining unchanged).

Request Parameters

Name	Mandatory	Description
dateFrom	N Default = today – 2 years	A valid date in yyyy-mm-dd format e.g. 2018-07-30. Type: alphanumeric, ISO8601 format
dateTo	N Default = today	A valid date in yyyy-mm-dd format e.g. 2020-07-30. Type: alphanumeric, ISO8601 format
role	N Default = buy, sell	Values: 'buy', 'sell'. Multiple values are permitted. Type: alphanumeric array
merchantRef	N	Merchant trade reference. Max 30 characters Type: alphanumeric
lwin	N	A valid LWIN7/11/16/18 code Type: integer, 7-digits (LWIN7), 11-digits (LWIN11), 16-digits (LWIN16), 18-digits (LWIN18)
vintage	N	If LWIN code is provided, then this parameter is ignored Type: 4-digit integer
packSize	N	If LWIN code is provided, then this parameter is ignored Type: 2-digit integer

bottleSize	N	If LWIN code is provided, then this parameter is ignored The values must be in ml (millilitres) expressed as a 5-digit term e.g. 75cl = 00750. Type: 5-digit integer
contractType	N Default = sib, sep, x	Values: 'sib', 'sep', 'x'. Multiple values are permitted. Type: alphanumeric array
paid	N	Type: Boolean 'true' or 'false'

Sample Request Body

JSON Request

```
{
  "yourTradesLists": {
    "dateFrom": "2020-01-01",
    "dateTo": "2020-06-01",
    "role": ["sell", "buy"],
    "contractType": ["SIB", "X"]
  }
}
```

XML Request

```
<yourTradesLists>
  <yourTradesList>
    <dateFrom>2020-01-01</dateFrom>
    <lwin>1011247</lwin>
    <merchantRef>P0</merchantRef>
    <paid>>false</paid>
  </yourTradesList>
</yourTradesLists>
```

Sample Response Body

The Positions service will respond with HTTP Code 200 OK in a successful response.

Response parameters

Name	Description
lxTradeNo	Type: integer
tradeDate	The timestamp of when the trade took place. Type: date (epoch time if JSON)
tradeStatus	Traded, deleted Type: alphanumeric
orderGUID	The Liv-ex order identifier. Type: 128-bit alphanumeric
lwin	LWIN18 Type: alphanumeric

lwinName	Vintage-specific LWIN wine name description e.g. 'Chateau Grand-Puy-Lacoste 5eme Cru Classe, Pauillac' Type: alphanumeric
lwinRegion	Region of origin (where applicable, specific to local laws). Type: alphanumeric
vintage	Type: alphanumeric
packSize	The LWIN pack size description e.g. '06' Type: alphanumeric
bottleSize	The LWIN bottle size description e.g. '00750' Type: alphanumeric
role	Buy, sell Type: alphanumeric
merchantRef	Type: alphanumeric
contractType	SIB, SEP, X Type: alphanumeric
quantity	Number of units traded Type: integer
currency	Either EUR, GBP or USD. The prices that follow will all be expressed in this currency. Type: alphanumeric
bottlePrice	Type: double
unitPrice	Type: double
totalUnitPrice	totalUnitPrice = quantity x unitPrice Type: double
commissionRate	Type: alphanumeric
commissionAmount	Type: double
settlementFee	Type: double
totalNetAmount	When role = buy: totalNetAmount = totalUnitPrice + settlementFee + comissionAmount When role = sell: totalNetAmount = totalUnitPrice - settlementFee - comissionAmount Type: double
paid	Type: Boolean true/false
prepaid	Type: Boolean true/false
invoiceNo	<i>null if not invoiced or where role = sell</i> Type: integer

JSON Response

The response is sent per request.

```
{
  "status": "OK",
  "statusCode": "200",
  "message": "Request completed successfully",
  "internalErrorCode": "R001",
  "apiInfo": {
    "version": "1.0",
    "timestamp": 1607596411238,
    "provider": "Liv-ex"
  },
  "pageInfo": {
    "totalResults": 2,
    "limit": 50,
    "offset": 1
  },
  "yourTradesList": [
    {
      "lxTradeNo": 225678,
      "tradeDate": 1591040328000,
      "tradeStatus": "traded",
      "orderGUID": "e59e46c3-662-c95607ce1f9b",
      "lwin": "101317920130600750",
      "lwinName": "Chateau La Mission Haut-Brion Cru Classe, Pessac-Leognan",
      "lwinRegion": "Bordeaux",
      "vintage": "2013",
      "packSize": "06",
      "bottleSize": "00750",
      "role": "Buy",
      "merchantRef": "BIDXX",
      "contractType": "SIB",
      "quantity": 1,
      "currency": "GBP",
      "bottlePrice": 103.0,
      "unitPrice": 618.0,
      "totalUnitPrice": 618.0,
      "commissionRate": "2%",
      "commissionAmount": 12.36,
      "settlementFee": 4.0,
      "totalNetAmount": 634.36,
      "paid": true,
      "prepaid": false,
      "invoiceNo": 319467,
      "errors": null
    },
    {
      "lxTradeNo": 2412336,
      "tradeDate": 1606985787000,
      "tradeStatus": "traded",
      "orderGUID": "6903-4683-9a18-a01e3d4b3781",
      "lwin": "100946620081200750",
      "lwinName": "Duclot, Collection Assortment Case",
      "lwinRegion": "Bordeaux",
      "vintage": "2008",
      "packSize": "12",
      "bottleSize": "00750",
      "role": "Sell",
      "merchantRef": "PO",
      "contractType": "X",
      "quantity": 3,
      "currency": "GBP",
      "bottlePrice": 58.33,
      "unitPrice": 700.0,
      "totalUnitPrice": 2100.0,
      "commissionRate": "2.00%",
      "commissionAmount": 42.0,
      "settlementFee": 12.0,
    }
  ]
}
```



```

        "totalNetAmount": 2046.0,
        "paid": false,
        "prepaid": null,
        "invoiceNo": null,
        "errors": null
    }
  ],
  "errors": null
}

```

Invalid JSON response

```

{
  "status": "Bad Request",
  "statusCode": "400",
  "message": "Request was unsuccessful",
  "internalErrorCode": "R000",
  "apiInfo": {
    "version": "1.0",
    "timestamp": 1607596673940,
    "provider": "Liv-ex"
  },
  "pageInfo": null,
  "yourTradesList": null,
  "errors": {
    "error": [
      {
        "code": "V158",
        "message": "Invalid role. Possible values are 'buy' or 'sell'."
      }
    ]
  }
}

```

XML Response

The response is sent per request.

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<yourTradesList>
  <Status>OK</Status>
  <HttpCode>200</HttpCode>
  <Message>Request completed successfully</Message>
  <InternalErrorCode>R001</InternalErrorCode>
  <ApiInfo>
    <Version>1.0</Version>
    <Timestamp>2020-12-10T11:00:47.804Z</Timestamp>
    <Provider>Liv-ex</Provider>
  </ApiInfo>
  <pageInfo>
    <totalResults>527</totalResults>
    <limit>1</limit>
    <offset>1</offset>
  </pageInfo>
  <yourTradesList>
    <lxTradeNo>2412436</lxTradeNo>
    <tradeDate>2020-12-03T08:56:27Z</tradeDate>
    <tradeStatus>traded</tradeStatus>
    <orderGUID>80e5-5a7f36639cf6</orderGUID>
    <lwin>100946620081200750</lwin>
    <lwinName>Chateau Haut-Brion, Pessac-Leognan</lwinName>
    <lwinRegion>Bordeaux</lwinRegion>
    <vintage>2008</vintage>
    <packSize>12</packSize>
    <bottleSize>00750</bottleSize>
  </yourTradesList>
</yourTradesList>

```

```

<role>Buy</role>
<merchantRef>PO 1234</merchantRef>
<contractType>X</contractType>
<quantity>3</quantity>
<currency>EUR</currency>
<bottlePrice>64.63</bottlePrice>
<unitPrice>775.57</unitPrice>
<totalUnitPrice>2326.71</totalUnitPrice>
<commissionRate>2.00%</commissionRate>
<commissionAmount>46.53</commissionAmount>
<settlementFee>15.0</settlementFee>
<totalNetAmount>2388.24</totalNetAmount>
<paid>>false</paid>
<prepaid>>false</prepaid>
<invoiceNo xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
<errors xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
</yourTradesList>
</yourTradesList>

```

Invalid XML Response

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<yourTradesList>
  <Status>Bad Request</Status>
  <HttpCode>400</HttpCode>
  <Message>Request was unsuccessful</Message>
  <InternalErrorCode>R000</InternalErrorCode>
  <ApiInfo>
    <Version>1.0</Version>
    <Timestamp>2020-12-10T11:02:50.799Z</Timestamp>
    <Provider>Liv-ex</Provider>
  </ApiInfo>
  <errors>
    <error>
      <code>V006</code>
      <message>Invalid LWIN number.</message>
    </error>
  </errors>
</yourTradesList>

```

6. Response Codes

This section describes the response codes that will be returned by the Exchange Integration services.

Code	Message
R000	Request was unsuccessful
R001	Request completed successfully

6.1 Request validation error codes

Code	Message
V006	Invalid LWIN number.
V045	Please provide valid bottle size.

V046	Please provide valid pack size.
V077	Invalid / incorrect contractType: [%s]. Possible values can be 'sib' (Standard In Bond), 'sep' (Standard En Primeur) and 'x' (Special).
V102	Invalid / incorrect orderType: [<value>]. Possible values are 'bid', 'offer'.
V157	Wrong date format. Requested date should be a valid date in 'yyyy-MM-dd' format
V158	Invalid role. Possible values are 'buy' or 'sell'.
V159	Invalid paid value. Possible values are 'true' or 'false'.

6.2 HTTP Status codes

HTTP defines a bunch of meaningful status codes that can be returned from our API. These can be leveraged to help our API Merchants/consumers route their responses accordingly:

Code	Message
200 OK	Response to a successful GET, POST, PUT, DELETE. Can also be used for a POST that doesn't result in a creation.
201 Created	Response to a POST that results in a creation.
202 Accepted	The request has been accepted and will be processed later. It is a classic answer to asynchronous calls (for better UX or performances).
204 No Content	Response to a successful request that won't be returning a body (like a DELETE request)
400 Bad Request	The request is malformed, such as if the body does not parse
401 Unauthorized	When no and/or invalid authentication details are provided. Can also be used to trigger an auth popup if API is used from a browser
403 Forbidden	When authentication succeeded but authenticated user doesn't have access to the resource
404 Not Found	When a non-existent resource is requested
405 Method Not Allowed	When an HTTP method is being requested that isn't allowed for the authenticated user
406 Not Acceptable	Nothing matches the Accept-* Header of the request. As an example, you ask for an XML formatted resource, but it is only available as JSON.
409 Conflict	Indicates one or more supplied parameters are triggering a validation error. A relevant TR code should be returned in the response.
410 Gone	Indicates that the resource at this end point is no longer available. Useful as a blanket response for old API versions
415 Unsupported Media Type	If incorrect content type was provided as part of the request

422 Unprocessable Entity	Used for validation errors. Should be used if the server cannot process the entity, e.g. if an image cannot be formatted or mandatory fields are missing in the payload.
429 Too Many Requests	When a request is rejected due to rate limiting
500 Internal Server Error	The general catch-all error when the server-side throws an exception. The request may be correct, but an execution problem has been encountered at our end.